Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

On convergence of implicit Runge-Kutta methods for the incompressible Navier-Stokes equations with unsteady inflow

Yunzhu Cai [a], Jiawei Wan [b,c,*], Ahsan Kareem [b]

[a] College of Civil Engineering, Nanjing Tech University, China
[b] Nathaz Modeling Laboratory, University of Notre Dame, United States
[c] China Energy Science and Technology Research Institute Co. Ltd., China

ARTICLE INFO

Dataset link: Replication code and data: On convergence of implicit Runge-Kutta methods for the incompressible Navier-Stokes equations with unsteady inflow (Original data)

Keywords:
Incompressible Navier-Stokes equations
Unsteady inflow
Differential-algebraic equations
Implicit Runge-Kutta methods
Order of convergence

ABSTRACT

This study investigates the convergence properties of implicit Runge-Kutta (IRK) methods when applied to the temporal solution of incompressible Navier-Stokes (N-S) equations with unsteady inflow. Owing to the differential-algebraic nature of spatially discretized N-S equations, conventional IRK methods may experience a significant order reduction while requiring exact satisfaction of the divergence-free constraint on the velocity field. Notably, the enhanced performance achieved through modified IRK techniques, such as projected Runge-Kutta methods and specialized Runge-Kutta methods, is confined to Runge-Kutta coefficients with specific attributes. In response to these limitations, this paper proposes a perturbed IRK scheme, modifying the intermediate stage equations of standard IRK methods by incorporating predefined perturbations, aiming to enhance convergence properties for the incompressible N-S equations accompanied by unsteady inflow. These perturbations within the scheme not only alleviate order reduction but also ensure exact enforcement of the divergence-free constraint. Moreover, the proposed scheme remains applicable in cases where the unsteady inflow is only available as discrete-time fields, rather than explicit functions of time. To demonstrate the efficiency of the proposed enhancement, an extensive analysis of the convergence properties for all considered IRK methods through a series of numerical experiments, is conducted.

1. Introduction

In the present study, we investigate the convergence properties of implicit Runge-Kutta (IRK) methods when applied to the temporal solution of incompressible Navier-Stokes (N-S) equations with unsteady inflow. Our interest in this topic is primarily motivated by the increasing prevalence of large-eddy simulations and direct numerical simulations for flow problems subjected to turbulent inflow in recent years [1]. Within the constraints of a fixed computational time budget, there is a need to enhance the temporal solution accuracy and stability of the N-S equations by selecting an appropriate scheme, particularly considering the intense velocity and pressure fluctuations induced by turbulent inflow [2]. This aspect has not been extensively addressed in prior studies [3–9] related to turbulent inflow simulations, which predominantly rely on schemes available in publicly released computational fluid dynamics (CFD) codes.

[*] Corresponding author at: Nathaz Modeling Laboratory, University of Notre Dame, United States.
E-mail addresses: yunzhucai@njtech.edu.cn (Y. Cai), jwan1@nd.edu (J. Wan), kareem@nd.edu (A. Kareem).

https://doi.org/10.1016/j.jcp.2024.113627
Received 1 March 2024; Received in revised form 20 October 2024; Accepted 22 November 2024

Available online 3 December 2024

The N-S equations for incompressible Newtonian fluids can be expressed as follows

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \nu \nabla^2 \boldsymbol{u} - \nabla p, \tag{1.1a}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1.1b}$$

Here, t represents time, $\boldsymbol{u}$ denotes the two or three-dimensional fluid velocity, p signifies the kinematic pressure, and ν stands for the kinematic viscosity. Spatial discretization of eq. (1.1) using a finite difference or finite volume method [10,11] yields a semi-discrete system concerning $\boldsymbol{u}$ and p at discrete spatial points in the computational of interest, which writes

$$u'(t) = -C(u(t))u(t) + \nu D u(t) - G p(t) + r(t, v(t)), \tag{1.2a}$$

$$0 = M u(t) + N v(t). \tag{1.2b}$$

In eq. (1.2) and the subsequent discussion, $u(t) \in \mathbb{R}^m$ and $p(t) \in \mathbb{R}^n$ represent vectors containing the spatially discretized velocity and pressure fields, respectively; $v(t) \in \mathbb{R}^q$ is a vector comprising the discrete velocities specified at inflow boundaries; $C \in \mathbb{R}^{m\times m}$, $D \in \mathbb{R}^{m\times m}$, $G \in \mathbb{R}^{m\times n}$, and $M \in \mathbb{R}^{n\times m}$ denote the matrix representations of the convection, diffusion, gradient, and divergence operators concerning u or p, respectively; $N \in \mathbb{R}^{n\times q}$ denotes a matrix whose product with v gives the contribution of inflow to the divergence of u; $r(t, v) \in \mathbb{R}^m$ is a vector resulting from v, unsteady boundary conditions for p and forcing terms for the momentum equation. The nonlinearity in the semi-discrete N-S equations primarily stems from the dependency of C on u. For the finite volume discretization of the incompressible Navier-Stokes equations on collocated grids, a velocity-flux field $\phi(t)$, evaluated on cell faces, is commonly introduced in addition to u. The continuity equation is then reformulated in terms of ϕ rather than u or v. Equation (1.2) accommodates the semi-discrete system in this context by incorporating ϕ into u and v (see Appendix F for more details).

In the case of spatial discretization on stationary grids, matrices D, G, M, and N are constant. However, they become time-varying when dealing with moving grids, except for the discrete continuity equation involving ϕ, where M and N retain their constancy. It is important to note that, in this study, we focus on finite volume-based fluid solvers, assuming that M and N are constant while allowing D and G to be time-varying, aligning with our primary interest.

It is evident that eq. (1.2) characterizes an index-2 system of differential-algebraic equations (DAEs), wherein u and p can be considered as the differential and algebraic components, respectively, according to the DAE literature [12,13]. This system can be further transformed into index-1 DAEs by substituting eq. (1.2a) into the time-differentiated form of eq. (1.2b). Thus, eq. (1.2b) is replaced by a Poisson equation concerning p that writes

$$M G p(t) = -M C(u(t))u(t) + \nu M D u(t) + M r(t) + N v'(t). \tag{1.3}$$

However, solving the semi-discrete incompressible Navier-Stokes equations as index-1 DAEs is generally discouraged in CFD. This is because the continuity equation (1.2b) is not enforced in conventional schemes used for solving eqs. (1.2a) and (1.3). Given that the numerical solution deviates from the exact solution of the system, this approach would prevent the exact enforcement of the divergence-free constraint. Introducing the notation

$$f(t, u, p) = -C(u)u + \nu D u - G p + r(t, v(t)), \quad g(t, u) = M u + N v(t), \tag{1.4}$$

eq. (1.2) can be simplified as a semi-explicit index-2 system

$$u' = f(t, u, p), \tag{1.5a}$$

$$0 = g(t, u). \tag{1.5b}$$

with $f(t, u, p) \in \mathbb{R}^m$ and $g(t, u) \in \mathbb{R}^n$ satisfying the following three hypotheses:

H1: $f(t, u, p)$ is sufficiently differentiable in u, while $g(t, u)$ is sufficiently differentiable in t.
H2: $g(t, u)$ is linear in u with the partial derivatives of g with respect to u and v (i.e., M and N) being constant.
H3: $f(t, u, p)$ is linear in p.

Treating both t and u as the differential components and p as the algebraic component, eq. (1.5) satisfying hypothesis H1 or both hypotheses H1 and H2 should be considered a general nonlinear semi-explicit index-2 system during the convergence analysis. In order that $g(t, u)$ is sufficiently differentiable in t, the boundary-normal component of the inflow velocity v (in other words, the mass flux across the inflow boundary) should be non-zero and sufficiently differentiable.

Numerical methods, which are at least second-order accurate for general semi-explicit index-2 DAEs, include multi-step methods [13], IRK methods [12,14,15], and half-explicit Runge-Kutta (HERK) methods [16,17]. A class of multi-step methods, known as the backward differential formulas (BDFs) [18–20], is very popular in DAEs for their capability of attaining the same order of convergence in DAEs as in ordinary differential equations (ODEs). In comparison, the standard IRK methods (also referred to as the direct approach in [12]), derived straightforwardly from conventional RK methods for ODEs, suffer from severe order reduction in index-2 DAEs. Some modified IRK methods, such as the projected Runge-Kutta method [21,22], specialized Runge-Kutta (SRK) methods [23], and partitioned Runge-Kutta (PRK) methods [24], have been proposed to overcome the order reduction phenomenon either by revising the original intermediate stage equations or introducing additional intermediate stage equations. These remedies either apply only to certain types of RK coefficients (details of which will be discussed later) or turn out to be too computationally demanding (in time or memory) to implement in CFD.

Early applications of RK methods in CFD [25–28] tended to march u in the time domain as if the semi-discrete incompressible N-S equations (1.2) defined a system of ODEs, and then corrected the resulting u to enforce the continuity equation (1.2b). It was not until early this decade [11,29–31] that the convergence of RK methods for the incompressible N-S equations was formally examined based on the existing literature on index-2 DAEs. Among all RK methods available, HERK methods seem to be a good choice since they only require solving a linear Poisson equation regarding p at each RK intermediate stage. It is suggested in [11] that a 3-stage HERK method, which can attain 3rd-order for velocity and 2nd-order for pressure, is an ideal choice that balances computational cost with sufficient accuracy. Compared to implicit Runge-Kutta (IRK) methods, half-explicit Runge-Kutta (HERK) methods generally require smaller time-step sizes to maintain stability, particularly when the semi-discrete Navier-Stokes systems exhibit stiffness. Furthermore, explicit RK methods can experience significant order reduction when applied to initial-boundary value problems (IBVPs) with time-dependent boundary conditions, which also applies to the semi-discrete Navier-Stokes system under consideration.

Within IRK methods, those with lower triangular coefficient matrices, i.e., diagonally-implicit Runge-Kutta (DIRK) methods [32] and explicit first-stage, singly diagonally-implicit Runge-Kutta (ESDIRK) methods [33,34], are much less computationally demanding than fully-implicit Runge-Kutta (FIRK) methods. DIRK methods were originally not encouraged for index-2 DAEs because their stage orders could only be first-order at most. After the idea of the so-called quasi-stage order [35] was proposed, DIRK methods were shown to be capable of attaining 3rd-order for differential components and 2nd-order for algebraic components with a stage number of no less than 3 [36]. A similar issue exists for ESDIRK methods, and it was not until quite recently in [37] that 3rd- and 4th-order ESDIRK methods were formally established for index-2 DAEs. Compared to DIRK and ESDIRK methods, FIRK methods (such as Gauss, Radau IA, and Radau IIA methods) usually attain higher orders of convergence given the same stage-number. However, they require the simultaneous solution of all implicit equations regarding the differential and algebraic components at RK intermediate stages, which severely hinders their applications in engineering practice. When applied to the semi-discrete N-S system (1.2), a noteworthy issue existing in all IRK methods is that it seems impossible to attain the same order of convergence for both u and p without solving the Poisson equation (1.3) at the end of each time-step [11]. Meanwhile, it is in many cases difficult to acquire the time-derivative of v required by the solution of eq. (1.3).

Unlike the dominating popularity of the BDFs in multi-step methods, we find it difficult to choose a particular IRK method that balances accuracy, efficiency, complexity, and stability from all methods available. This could be a possible factor leading to the fact that few attempts had been made to solve the incompressible N-S equations with turbulent inflow using IRK methods. Another possible factor is that turbulent inflow velocity fields are commonly acquired from a precursor simulation or a synthetic approach in practice. It is quite uncommon, sometimes troublesome, for either of these two approaches to generate turbulent velocities at specific non-uniform time instants, which are required by the application of most RK methods. To this end, we first comprehensively summarize and compare the convergence of various IRK methods for the incompressible N-S equations, and then explore a possibility of any improvement by revising the original intermediate stage equations of IRK methods.

This paper is organized as follows. Section 2 introduces the application and convergence of various IRK methods for the semi-discrete incompressible N-S equations (1.2). A remedy of improving the convergence of conventional IRK methods for eq. (1.2) is presented in Section 3. Comments on the linear and nonlinear stability of RK methods are given in Section 4. Section 5 examines the considered IRK methods and the proposed enhancement via three numerical experiments. Finally, Section 6 concludes the paper with a summary of the findings and limitations.

2. IRK methods for the incompressible N-S equations

This section briefly reviews the formulation and convergence properties of various IRK methods applied to the semi-discrete incompressible N-S equations (1.2). In addition to standard IRK methods, SRK methods are incorporated in the discussion. Projected Runge-Kutta methods are not considered, as their convergence results for index-2 DAEs offer no improvement over SRK methods while necessitating an additional projection step. Although solving the semi-discrete incompressible N-S equations as index-1 DAEs is not recommended, their convergence properties are discussed in this section. Related order results can elucidate the extent of order reduction experienced by IRK methods with different types of RK coefficients and the potential for enhancement concerning these methods. Following eq. (1.3), the reduced-index system of (1.5) is expressed as

$$u'(t) = f(t,u,p), \tag{2.1a}$$

$$0 = Mu'(t) + Nv'(t). \tag{2.1b}$$

To ensure the solvability of the DAE system (1.5) or its equivalent form (2.1), the existence and boundedness of the inverse of MG are assumed. The initial values (u_0, p_0) of the system are presumed to satisfy the consistency condition, that is,

$$g(t_0,u_0) = 0, \quad Mu'(t_0) + Nv'(t_0) = 0. \tag{2.2}$$

2.1. IRK methods for the index-1 DAEs (2.1)

We begin with the index-1 system (2.1). An s-stage standard IRK method with a non-singular coefficient matrix approximates the exact solution of $u(t)$ at $t_{n+1} = t_n + h$ (where h represents the time-step size in this context) from an approximation u_n at t_n via the formula

$$u_{n+1} = u_n + h \sum_{i=1}^{s} b_i f(t_n + c_i h, U_{ni}, P_{ni}), \tag{2.3a}$$

where the values of (U_{ni}, P_{ni}) at the i-th intermediate stage (for $i = 1, \cdots, s$) are obtained by solving a system of nonlinear equations, which takes the form

$$U_{ni} = u_n + h \sum_{j=1}^{s} a_{ij} f(t_n + c_j h, U_{nj}, P_{nj}), \tag{2.3b}$$

$$0 = M f(t_n + c_i h, U_{ni}, P_{ni}) + N v'(t_n + c_i h). \tag{2.3c}$$

Throughout this paper, the subscript n in variables such as u_n and p_n represents the values of corresponding variables at t_n, and the subscripts ni in variables like U_{ni} and P_{ni} represent the values of corresponding variables at the i-th RK intermediate stage when the numerical solution progresses from t_n to t_{n+1}. We use the notation a_{ij}, b_i, and c_i for the coefficients of a RK method and $[\omega_{ij}]$ for the inverse of the matrix $[a_{ij}]$ (if it exists). For simplicity, we denote $A \in \mathbb{R}^{s\times s}$ as the matrix of $[a_{ij}]$, $b \in \mathbb{R}^s$ as the vector of b_i, $c \in \mathbb{R}^s$ as the vector of c_i, and t_{ni} as $t_n + c_i h$. It is also convenient to use the notation $\tilde{b} = (b_2, \cdots, b_s)$ and $\tilde{c} = (c_2, \cdots, c_s)$ during the discussion of IRK methods with $a_{1i} = 0$ for all i. Finally, it is crucial to acknowledge that the IRK methods considered in this study can be classified into two major categories, according to the properties of their coefficients:

Type I: A is non-singular, encompassing Gauss, Radau IA, Radau IIA, Lobatto IIIC, and DIRK methods;

Type II: the conditions $a_{1i} = 0$ and $a_{si} = b_i$ hold for all i, concurrently with the non-singularity of the sub-matrix $\tilde{A} = [a_{ij}]_{i,j\geq 2}$ of A, as exemplified by Lobatto IIIA and ESDIRK methods.

The stability function [12,33] associated RK coefficients of type I and II satisfies

$$R(\infty) = \begin{cases} 1 - b^T A^{-1}\mathbb{1} & \text{for type I coefficients,} \\ 1 - \tilde{b}^T \tilde{A}^{-2}\tilde{c} & \text{for type II coefficients,} \end{cases} \tag{2.4}$$

where $\mathbb{1}$ denotes a vector of 1.

To distinguish between the standard IRK methods for the equivalent index-2 system (1.5), which will be introduced later, we refer to eq. (2.3) as the IRK-DAE1 scheme hereafter. We also employ the notation $B(\eta)$, $C(\zeta)$, and $D(\xi)$, following [12,38], to represent the conditions of

$$\begin{aligned} B(\eta) &: \sum_{i=1}^{s} b_i c_i^{k-1} = \frac{1}{k} \text{ for } k = 1, \ldots, \eta, \\ C(\zeta) &: \sum_{j=1}^{s} a_{ij} c_j^{k-1} = \frac{c_i^k}{k} \text{ for } k = 1, \ldots, \zeta \text{ and all } i, \\ D(\xi) &: \sum_{i=1}^{s} b_i c_i^{k-1} a_{ij} = \frac{b_j}{k}(1 - c_j^k) \text{ for } k = 1, \ldots, \xi \text{ and all } j. \end{aligned} \tag{2.5}$$

Once u_{n+1} is obtained from eq. (2.3), the exact solution of $p(t)$ at t_{n+1}, for IRK methods of type I and II, can be approximated with the solution of

$$M f(t_{n+1}, u_{n+1}, p_{n+1}) + N v'(t_{n+1}) = 0. \tag{2.6}$$

For IRK methods of type I, p_{n+1} can also be obtained from

$$p_{n+1} = p_n + \sum_{i,j=1}^{s} b_i \omega_{ij}(P_{nj} - p_n). \tag{2.7}$$

When the RK coefficients satisfy the condition $a_{si} = b_i$ for all i, or in other words, the IRK method is stiffly accurate, we obtain $u_{n+1} = U_{ns}$ from eq. (2.3b), and consequently, the P_{ns} satisfying eq. (2.3c) (with $i = s$) is identical to the p_{n+1} given by eq. (2.6). Meanwhile, eq. (2.7) reduces to $p_{n+1} = P_{ni}$ if $a_{si} = b_i$. Thus, for stiffly accurate methods, the two formulas eqs. (2.6) and (2.7) are equivalent. For non-stiffly accurate methods, the two formulas provide different approximations for p_{n+1}, and the corresponding convergence results also differ. It is important to note that, compared to eq. (2.7), the determination of p_{n+1} using eq. (2.6) results in n additional algebraic equations to be solved within a numerical step.

The convergence of the IRK-DAE1 scheme is summarized in Appendix A. From eq. (A.1), one can easily verify that the u-component in eq. (2.3) fails to satisfy eq. (1.5b) exactly. Due to the accumulation of the global error for the u-component as the step-number n continuously increases, the residual error of $g(t_n, u_n)$ would become more and more significant. This may cause non-physical results and threaten the numerical stability of the solution when the IRK-DAE1 scheme (2.3) is applied to the solution of the incompressible N-S equations.

2.2. IRK methods for the index-2 DAEs (1.5)

For the index-2 system (1.5), an s-stage standard IRK method can be expressed as

$$u_{n+1} = u_n + h \sum_{i=1}^{s} b_i f(t_{ni}, U_{ni}, P_{ni}), \tag{2.8a}$$

$$U_{ni} = u_n + h \sum_{j=1}^{s} a_{ij} f(t_{nj}, U_{nj}, P_{nj}), \tag{2.8b}$$

$$0 = g(t_{ni}, U_{ni}), \tag{2.8c}$$

which will be referred to as the IRK-DAE2 scheme in the following context. Similar to the index-1 system (2.1), the approximation p_{n+1} can be obtained either from eq. (2.6) or eq. (2.7) for IRK methods of type I, where two formulas give different p_{n+1} and attain different orders for the global error of the p-component. For IRK methods of type II, eq. (2.6) can still be used but a more economic option is to utilize $p_{n+1} = P_{ns}$, and their convergence results also differ. As for the enforcement of the algebraic constraint, we have $u_{n+1} = U_{ns}$ and consequently $g(t_{n+1}, u_{n+1}) = 0$ for the IRK-DAE2 scheme when the IRK method is stiffly accurate. For non-stiffly accurate IRK methods (of type I), the residual error of $g(t_{n+1}, u_{n+1})$ is bounded and has the same order as that of the local error of the method for the u-component. The convergence of the IRK-DAE2 scheme is summarized in Appendix B.

2.3. SRK methods for the index-2 DAEs (1.5)

In an s-stage SRK method, the values of (U_{ni}, P_{ni}) at the i-th intermediate stage are obtained from the solution of the following system

$$u_{n+1} = u_n + h \sum_{i=1}^{s} b_i f(t_{ni}, U_{ni}, P_{ni}), \tag{2.9a}$$

$$U_{ni} = u_n + h \sum_{j=1}^{s} a_{ij} f(t_{nj}, U_{nj}, P_{nj}), \tag{2.9b}$$

$$0 = \sum_{j=1}^{s} \tilde{\omega}_{ij} g(t_{nj,} U_{nj}) + \tilde{\omega}_{i,s+1} g(t_{n+1}, u_{n+1}). \tag{2.9c}$$

Equation (2.9) will be referred to as the SRK-DAE2 scheme in the following context. In contrast to the IRK-DAE1 or IRK-DAE2 scheme, u_{n+1} cannot be explicitly evaluated from eq. (2.9a), unless $u_{n+1} = U_{ns}$, due to its presence in the intermediate stage equation of eq. (2.9c). In other words, u_{n+1} effectively becomes an unknown variable required to be solved for in RK intermediate stages (similar to U_{ni}) for non-stiffly accurate methods, which makes the SRK-DAE2 scheme more computationally expensive than the previous two schemes. Again, p_{n+1} can be computed either with eq. (2.6) or eq. (2.7), and the corresponding convergence results differ. The coefficients $\tilde{\omega}_{ij}$, additionally introduced in SRK methods, define a matrix Ω of the dimension $s \times (s+1)$. It is required for SRK methods that $\Omega\alpha \in \mathbb{R}^{s\times s}$ be non-singular, with the notation α defined as

$$\alpha = \begin{pmatrix} A \\ b^T \end{pmatrix}. \tag{2.10}$$

For the SRK-DAE2 scheme (2.9) to attain its optimal order results, the matrix Ω, according to the work of [23], can be chosen as

$$\Omega = \begin{pmatrix} 0_s^T & 1 \\ b^T & 0 \\ \vdots & \vdots \\ b^T C^{s-2} & 0 \end{pmatrix} \in \mathbb{R}^{s\times(s+1)}, \tag{2.11}$$

which satisfies $\Omega\alpha = I_s$ (the notation I denotes the identity matrix and the subscript s in I represents its dimension). Note that the use of Ω satisfying $\Omega\alpha = I_s$ brings a favorable property concerning the enforcement of the algebraic constraint. To see this, we rewrite eq. (2.9c) as

$$(\Omega \otimes I_n) \begin{pmatrix} g(t_{n1}, Y_{n1}) \\ \vdots \\ g(t_{ns}, U_{ns}) \\ g(t_{n+1}, u_{n+1}) \end{pmatrix} = 0, \tag{2.12}$$

where $\otimes$ denotes the tensor matrix product notation. Multiplying eq. (2.10) to both sides of the above equation, we have $g(t_{ni}, Y_{ni}) = g(t_{n+1}, u_{n+1}) = 0$ in SRK methods. This also indicates that the SRK-DAE2 scheme is actually identical to the IRK-DAE2 scheme for stiffly accurate methods. Finally, we note that the SRK-DAE2 scheme requires the non-singularity of A and is therefore not compatible with IRK methods of type II. The convergence of the SRK-DAE2 scheme is summarized in Appendix C.

Table 2.1
Properties concerning the coefficients of some classic IRK methods and their orders (denoted by κ) for ODEs.

method	stages	$B(\eta)$	$C(\zeta)$	$D(\xi)$	κ	type	other properties
Gauss	s	$2s$	s	s	$2s$	I	$R(\infty)=1$ (or -1) if s is even (or odd)
Radau IA	s	$2s-1$	$s-1$	s	$2s-1$	I	$c_1=0$
Radau IIA	s	$2s-1$	s	$s-1$	$2s-1$	I	stiffly accurate
Lobatto IIIA	s	$2s-2$	s	$s-2$	$2s-2$	II	$R(\infty)=-1$ (or 1) if s is even (or odd)
Lobatto IIIC	s	$2s-2$	$s-1$	$s-1$	$2s-2$	I	$c_1=0$ and stiffly accurate
SDIRK [32]	3	3	1	0	3	I	stiffly accurate
SDIRK [35]	4	3	1	0	3	I	2nd quasi-stage order and stiffly accurate
SDIRK [13]	5	4	1	0	4	I	stiffly accurate
ESDIRK [37]	3	2	2	0	2	II	$R(\infty)=0$ and stiffly accurate
ESDIRK [39]	4	3	2	0	3	II	$R(\infty)=0$ and stiffly accurate
ESDIRK [36]	6	4	2	0	4	II	$R(\infty)=0$, stiffly accurate and $\tilde{b}^T\tilde{A}^{-2}\tilde{c}^3=3$

Table 2.2
Orders of the global errors for the IRK-DAE1, IRK-DAE2 and SRK-DAE2 schemes with the p-component computed with eq. (2.7) or $p_{n+1}=P_{ns}$.

method	stages	IRK-DAE1 (2.3)		IRK-DAE2 (2.8)		SRK-DAE2 (2.9)	
		u	p	u	p	u	p
Gauss	$s\begin{cases}\text{odd}\\\text{even}\end{cases}$	$2s$	$\begin{cases}s+1\\s\end{cases}$	$\begin{cases}s+1\\s\end{cases}$	$\begin{cases}s-1\\s-2\end{cases}$	$2s$	$\begin{cases}s+1\\s\end{cases}$
Radau IA	s	$2s-1$	s	s	$s-1$	$2s-1$	s
Radau IIA	s	$2s-1$	$2s-1$	$2s-1$	s	$2s-1$	s
Lobatto IIIA	$s\begin{cases}\text{odd}\\\text{even}\end{cases}$	$2s-2$	$2s-2$	$2s-2$	$\begin{cases}s-1\\s\end{cases}$	-	-
Lobatto IIIC	s	$2s-2$	$2s-2$	$2s-2$	$s-1$	$2s-2$	$s-1$
SDIRK [32]	3	3	3	2	1	2	1
SDIRK [35]	4	3	3	3	2	3	2
SDIRK [13]	5	4	4	2	1	2	1
ESDIRK [37]	3	2	2	2	2	-	-
ESDIRK [39]	4	3	3	3	2	-	-
ESDIRK [36]	6	4	4	4	3	-	-

2.4. Comparison of the IRK-DAE1, IRK-DAE2 and SRK-DAE2 schemes

To compare the convergence of the three RK schemes eqs. (2.3), (2.8) and (2.9) from a more objective perspective, we first consider the case in which the p-components in the three schemes are all computed with eq. (2.7). In this case, the IRK-DAE1 and IRK-DAE2 schemes require the solution of $(m+n)\times s$ implicit equations at RK intermediate stages, whereas the SRK-DAE2 scheme demands the additional solution of m implicit equations for non-stiffly accurate methods. In the case of stiffly accurate methods, the IRK-DAE2 and SRK-DAE2 schemes are equivalent. Moreover, the intermediate stage equations of the IRK-DAE1 or IRK-DAE2 scheme can be solved in s successive stages with $(m+n)$ implicit equations to be solved at each intermediate stage when A is lower triangular. In comparison, the SRK-DAE2 scheme still requires the simultaneous solution of all implicit equations of intermediate stages, unless the coefficients are stiffly accurate.

According to the convergence results summarized in Appendix A, B and C, we list the orders of the three considered schemes for some classic RK methods (Table 2.1), including Gauss, Radau IA, Radau IIA, Lobatto IIIA, Lobatto IIIC, SDIRK, and ESDIRK methods in Table 2.2. Within the three SDIRK methods listed in the table, the 3-stage and 5-stage ones are originally conceived for ODEs, while the 4-stage method is specially designed for index-2 DAEs with a quasi-stage-order of 2. The three ESDIRK methods considered in the table are also established for DAEs of higher indices. When implemented with the IRK-DAE2 scheme, they can attain 2nd-order for algebraic components at least. Also, note that, except for the Gauss and Radau IA methods, all remaining methods in the table are stiffly accurate.

Among the three IRK schemes, the IRK-DAE1 scheme attains the highest orders of convergence for all methods. Apart from the 3-stage and 5-stage SDIRK method, the SRK-DAE2 scheme achieves the same orders as those of the IRK-DAE1 scheme for the global error of the u-component. In comparison, the IRK-DAE2 scheme can only obtain identical order results in the case of Radau IIA, Lobatto IIIA, and Lobatto IIIC methods, as well as the DIRK and ESDIRK method designed for index-2 DAEs. As for the global error of the p-component, the SRK-DAE2 and IRK-DAE2 schemes obtain identical orders, lower than those of the IRK-DAE1 scheme, for all the stiffly accurate methods in Table 2.2. For the Gauss and Radau IA methods, the IRK-DAE1 and SRK-DAE2 schemes share the same orders for the global error of the p-component.

When judging from the convergence properties alone, the IRK-DAE1 scheme should be our first choice, considering it attains the best convergence while using the least computational effort for each of the methods listed in Table 2.1. However, the numerical solution given by it fails to satisfy the constraint of eq. (1.5b). For Gauss and Radau IA methods, the SRK-DAE2 scheme seems to be an appealing option since it catches up with the IRK-DAE1 scheme in order results with the algebraic constraint eq. (1.5b) being satisfied exactly. Comparing the order results between the IRK-DAE1 and IRK-DAE2 schemes, significant order reduction can be noticed, especially for the 3-stage and 5-stage SDIRK method fulfilling $C(1)$ only.

Next, we consider the case in which the p-components in the three IRK schemes are all computed with eq. (2.6). Recall that for stiffly accurate methods, eqs. (2.6) and (2.7) are equivalent in the IRK-DAE1 scheme, indicating that no additional implicit equations need to be solved. Apart from this special case, all of the three schemes are required to solve n additional implicit equations at the end of each numerical step due to the use of eq. (2.6). When the order of the p-component is leveled to that of the u-component, the SRK-DAE2 scheme is a better choice than the IRK-DAE2 scheme, in terms of the Gauss and Radau IA methods, considering its superconvergence for both u- and p-components. The IRK-DAE2 and SRK-DAE2 schemes also catch up with the convergence orders of the IRK-DAE1 scheme in the case of Radau IA and Lobatto IIIC methods, as well as the 4-stage SDIRK method and the three ESDIRK methods. In these methods, both schemes are more computationally demanding than the IRK-DAE1 scheme, but their solutions can satisfy the continuity equation exactly.

2.5. A note on order reduction from space–time errors

Another mechanism for order reduction in Runge-Kutta (RK) methods is related to unsteady boundary conditions, in addition to the previously discussed reduction concerning lower convergence orders for index-2 DAEs compared to ODEs and index-1 DAEs. This phenomenon was first observed when applying ERK methods to initial-boundary value problems (IBVPs) expressed as partial differential equations with time-dependent boundary conditions [40]. In such problems, where both the mesh and time step are refined simultaneously, the numerical solution is expected to converge to the exact solution of the IBVPs. To estimate the space-time error, defined as the difference between the numerical solution and the exact solution of the IBVPs (rather than the semi-discrete IBVPs), it is necessary for the spatial-derivative operator, applied to the spatial discretization of the IBVPs, to return the time derivative of the dependent variable at both the interior and boundary nodes. This requires a semi-discrete equation at the boundary. Consequently, it is feasible to solve the semi-discrete IBVPs using RK methods without specifying intermediate boundary values, as they can be obtained by integrating the semi-discrete equation at the boundary using RK methods. This approach, while maintaining the classical convergence of RK methods, significantly restricts the allowable time step for stability. In contrast, the conventional approach imposes exact boundary values at intermediate stages based on the time-dependent boundary conditions. This results in a perturbed RK scheme where the perturbations, introduced in the fully discrete equation at the boundary, correspond to the difference between the exact boundary values and those approximated by the RK method. These perturbations are of the same order as the stage-order of the RK methods. As a result, order reduction occurs in RK methods with stage-order lower than the classical order, particularly affecting ERK methods, which have a stage-order of no more than one.

To alleviate this order reduction, constructing IRK methods with higher stage-order is the most straightforward solution, though it requires fully implicit RK coefficients. Alternative approaches have been explored, and two main strategies are available. The first involves designing RK coefficients that satisfy additional conditions, such as weak stage-order [41–43], which are less restrictive than stage-order and can be applied directly to DAEs without compromising convergence. The second approach is to impose perturbed boundary values during the solution process [40,44–46]. These perturbed values exhibit the same truncation errors as those obtained when no intermediate boundary conditions are imposed. However, the effectiveness of this method for DAEs, particularly for the incompressible Navier–Stokes equations, remains unvalidated. The primary focus of the current study is the convergence of global errors, defined as the difference between the numerical solution and the exact solution of the semi-discrete incompressible Navier-Stokes equations. Further investigation into this issue is reserved for future research.

3. Improving the convergence of standard IRK methods

As discussed in the preceding section, the IRK-DAE1 scheme demonstrates greater efficiency than both the IRK-DAE2 and SRK-DAE2 schemes in achieving equivalent or higher orders of convergence for the DAEs outlined in eq. (1.5). Nonetheless, the numerical solution provided by the IRK-DAE1 scheme does not satisfy the algebraic constraint. In contrast, the SRK-DAE2 scheme enforces the algebraic constraint precisely, albeit at the expense of increased computational effort, and achieves similar convergence results when employing the Gauss and Radau IA methods. For DIRK methods, which exhibit a notable advantage in computational efficiency compared to FIRK methods, both the IRK-DAE2 and SRK-DAE2 schemes exhibit inferior performance relative to the IRK-DAE1 scheme. Consequently, the question arises: Is it feasible to achieve order results comparable to the IRK-DAE1 scheme while utilizing roughly the same computational effort and satisfying the algebraic constraint of eq. (1.5b) exactly? The answer is affirmative, by imposing perturbations to intermediate boundary values of the velocity in the IRK-DAE1 scheme.

Perturbed schemes, which involve adding perturbations to the original formulations of multi-step methods or RK methods applied to ODEs or DAEs, are frequently used in the analysis of local errors and global convergence. Depending on the specific objective, the perturbations in different schemes can represent round-off errors, residual errors from the iterative solution of nonlinear discretized equations, or local truncation errors [12] when the exact solution is substituted into the unperturbed scheme. When perturbations are viewed as local truncation errors, where the unperturbed scheme holds for the exact solution, the local error of the unperturbed scheme can be estimated based on the magnitude of the perturbations. When perturbations are introduced in a scheme starting from a different initial value, analyzing the difference between the numerical solutions of the perturbed and unperturbed schemes aids in evaluating the convergence of the original scheme [12].

Beyond their essential role in error and convergence analysis, perturbations with specific values can even enhance the convergence of a method for certain problems. To mitigate the order reduction encountered by RK methods when solving IBVPs with time-dependent boundary conditions, various approaches have been proposed. For ERK methods, perturbing intermediate boundary values with prescribed modifications was suggested in [40,44,45], while similar remedies were developed for IRK methods in [46]. These perturbed boundary values are designed to produce the same local truncation errors as the RK method would when no intermediate boundary conditions are applied. For instance, in the work by [44], this is achieved for the classical 4-stage 4th-order RK method by deriving intermediate boundary values from the intermediate stage values of the same method applied to $\frac{\mathrm{d}^3 v}{\mathrm{d}t^3} = v'''(t)$.

In this study, prescribed perturbations are applied to the IRK-DAE1 scheme (2.3) to enforce the algebraic constraint in eq. (1.2b), while preserving the original convergence properties. Due to the fundamental differences between ODEs and DAEs, the impact of these perturbations on the convergence of the IRK-DAE1 scheme warrants separate investigation.

3.1. A perturbed IRK scheme for the system (1.5) and its convergence results

By regarding the boundary velocity $v(t)$ in the index-1 system (2.1) as an unknown variable to be solved for given $v'(t)$, rather than an explicit function, an s-stage IRK method can be expressed as follows

$$u_{n+1} = u_n + h\sum_{i=1}^{s} b_i U'_{ni}, \quad v_{n+1} = v_n + h\sum_{i=1}^{s} b_i v'(t_{ni}), \tag{3.1a}$$

$$U_{ni} = u_n + h\sum_{j=1}^{s} a_{ij} U'_{nj}, \quad V_{ni} = v_n + h\sum_{j=1}^{s} a_{ij} v'(t_{nj}), \tag{3.1b}$$

$$0 = MU'_{ni} + Nv'(t_{ni}), \tag{3.1c}$$

where U'_{ni} (for $i = 1, \cdots, s$) is defined as

$$U'_{ni} = -C(U_{ni})U_{ni} + \nu DU_{ni} - GP_{ni} + r(t_{ni}, V_{ni}), \tag{3.2}$$

and the initial condition for v is specified by $v_0 = v(t_0)$. In comparison to the IRK-DAE1 scheme, the boundary velocity values at t_{ni} and t_{n+1} are now approximated by V_{ni} and v_{n+1} in the scheme of eq. (3.1). It is straightforward to confirm that the convergence of IRK methods for index-1 DAE systems remains applicable to the scheme of eq. (3.1). Analogous to the IRK-DAE1 scheme, the solution u_{n+1} obtained by eq. (3.1) does not satisfy $g(t_{n+1}, u_{n+1}) = 0$. Additionally, the v_{n+1} in eq. (3.1) is not equal to $v(t_{n+1})$ due to errors stemming from RK approximations.

We now modify the equations in eq. (3.1) by considering a perturbed scheme, given as follows

$$u_{n+1} = u_n + h\sum_{i=1}^{s} b_i U'_{ni} + h\delta^u_{n+1}, \quad v_{n+1} = v_n + h\sum_{i=1}^{s} b_i v'(t_{ni}) + h\delta^v_{n+1}, \tag{3.3a}$$

$$U_{ni} = u_n + h\sum_{j=1}^{s} a_{ij} U'_{nj} + \delta^u_{ni}, \quad V_{ni} = v_n + h\sum_{j=1}^{s} a_{ij} v'(t_{nj}) + \delta^v_{ni}, \tag{3.3b}$$

$$0 = M\left(U'_{ni} + \theta^u_{ni}\right) + N\left(v'(t_{ni}) + \theta^v_{ni}\right), \tag{3.3c}$$

where δ^u_{n+1}, δ^v_{n+1}, δ^u_{ni}, δ^v_{ni}, θ^u_{ni}, and θ^v_{ni} represent perturbation terms that fulfill the constraints of

$$\delta^u_{ni} = h\sum_{j=1}^{s} a_{ij}\theta^u_{nj}, \tag{3.4a}$$

$$\delta^u_{n+1} = \sum_{i=1}^{s} b_i\theta^u_{ni}, \tag{3.4b}$$

$$\delta^v_{ni} = h\sum_{j=1}^{s} a_{ij}\theta^v_{nj}, \tag{3.4c}$$

$$\delta^v_{n+1} = \sum_{i=1}^{s} b_i\theta^v_{ni} = \frac{1}{h}\left(v(t_{n+1}) - v(t_n) - h\sum_{i=1}^{s} b_i v'(t_{ni})\right). \tag{3.4d}$$

Multiplying M on both sides of the first relation of eq. (3.3b), we obtain

$$MU_{ni} = Mu_n + h\sum_{j=1}^{s} a_{ij} MU'_{nj} + M\delta^u_{ni}. \tag{3.5}$$

Using eq. (3.3c) for the computation of U'_{ni} on the right hand side of eq. (3.5), this yields

$$MU_{ni} = Mu_n - h\sum_{j=1}^{s} a_{ij}\left[M\theta^u_{nj} + N\left(v'(t_{nj}) + \theta^v_{nj}\right)\right] + M\delta^u_{ni}. \tag{3.6}$$

Employing the second relation of eq. (3.3b) and $g(t_n, u_n) = Mu_n + Nv(t_n)$, eq. (3.6) can be rewritten as

$$MU_{ni} + NV_{ni} = g(t_n, u_n) + M\left(\delta^u_{ni} - h\sum_{j=1}^{s} a_{ij}\theta^u_{nj}\right) + N\left(\delta^v_{ni} - h\sum_{j=1}^{s} a_{ij}\theta^v_{nj}\right) + N\left(v_n - v(t_n)\right). \tag{3.7}$$

Based on the relations of eqs. (3.4a) and (3.4c) and assuming that $g(t_n, u_n) = 0$ and $v_n = v(t_n)$ in the scheme of eq. (3.3) (which will be verified later), the above equality can be further simplified as

$$MU_{ni} + NV_{ni} = 0. \tag{3.8}$$

Hereafter, we will refer to eq. (3.3) or its equivalent form with eq. (3.3c) replaced by eq. (3.8) as an implicit Runge-Kutta scheme with constrained perturbations (denoted by IRK-CP hereafter) for simplicity. The roles of these perturbations, how to determine them, and how they affect the convergence of the IRK-CP scheme (3.3) will be discussed successively.

To begin with, the perturbations δ^u_{ni} and θ^u_{ni} account for the residual error brought by the inexact solution of RK intermediate stages. In other words, they are treated as zeros when solving the algebraic system of eqs. (3.3b) and (3.3c). Apparently, such a system cannot be solved exactly due to the non-linearity in $C(U_{ni})$. The solution (U_{ni}, P_{ni}) obtained from a general non-linear equation solver can only satisfy eqs. (3.3b) and (3.3c) with residual errors existing in both equations. It should be noted that δ^u_{ni} and θ^u_{ni} do not satisfy eq. (3.4a) in general. However, this constraint can be satisfied given a proper solution algorithm, which will be addressed later.

The remaining perturbations are all prescribed items introduced in the IRK-CP scheme for specific purposes. More specifically, δ^v_{n+1} makes the boundary velocity given by the second relation of eq. (3.3a) identical to its exact value at t_{n+1} by canceling out the local error of RK approximations. Meanwhile, δ^u_{n+1}, δ^v_{ni}, and θ^v_{ni} together ensure the exact satisfaction of $g(t_{n+1}, u_{n+1}) = 0$. To verify this, we first consider IRK methods of type I and rewrite the first relation of eq. (3.3a) as

$$u_{n+1} = u_n + \sum_{i,j=1}^{s} b_i\omega_{ij}\left(U_{nj} - u_n - \delta^u_{nj}\right) + h\delta^u_{n+1}, \tag{3.9}$$

based on the first relation of eq. (3.3b) and the non-singularity of $[a_{ij}]$. Similar to eq. (3.5), multiplying M on both sides of eq. (3.9) yields

$$Mu_{n+1} = Mu_n + \sum_{i,j=1}^{s} b_i\omega_{ij}M\left(U_{nj} - u_n - \delta^u_{nj}\right) + hM\delta^u_{n+1}. \tag{3.10}$$

Substituting eq. (3.6) into eq. (3.10), replacing Mu_n with $g(t_n, u_n) - Nv(t_n)$ and similarly for Mu_{n+1}, we arrive at

$$g(t_{n+1}, u_{n+1}) = g(t_n, u_n) + hM\delta^u_{n+1} + N\left(v(t_{n+1}) - v(t_n) - h\sum_{i=1}^{s} b_i v'(t_{ni})\right) - h\sum_{i=1}^{s} b_i\left(M\theta^u_{ni} + N\theta^v_{ni}\right). \tag{3.11}$$

Based on eqs. (3.4b) and (3.4d), the right-hand side of eq. (3.11) equals zero if $g(t_n, u_n) = 0$. Considering the initial condition satisfies $g(t_0, u_0) = 0$, we deduce that $g(t_n, u_n) = 0$ and consequently $g(t_{n+1}, u_{n+1}) = 0$. For IRK methods of type II, it is easy to find $h\delta^u_{n+1} = \delta^u_{ns}$, $h\delta^v_{n+1} = \delta^v_{ns}$, $u_{n+1} = U_{ns}$ and consequently $g(t_{n+1}, u_{n+1}) = 0$, following from eqs. (3.4) and (3.8).

It is straightforward to tell eq. (3.8) happens to be the continuity equation regarding U_{ni} and V_{ni}. We therefore prefer to use eq. (3.8) instead of eq. (3.3c) when applying the scheme to the solution of incompressible N-S equations, considering the resemblance between the fully-discrete N-S system of eqs. (3.3b) and (3.8) and those from other temporal discretization schemes. Furthermore, the commonly used iterative solution methods for the fully-discrete N-S system, such as the well-known PISO algorithm, can ensure the exact enforcement of eq. (3.8). When eq. (3.8) holds exactly, the above-mentioned constraint concerning δ^u_{ni} and θ^u_{ni}, i.e., eq. (3.4a), is satisfied based on eqs. (3.7) and (3.4c).

In the following discussion, it is helpful to group similar perturbation terms together as one with the use of the notation

$$\delta_{n+1} = \begin{pmatrix} \delta^u_{n+1} \\ \delta^v_{n+1} \end{pmatrix}, \quad \delta_{ni} = \begin{pmatrix} \delta^u_{ni} \\ \delta^v_{ni} \end{pmatrix}, \quad \theta_{ni} = M\theta^u_{ni} + N\theta^v_{ni}, \tag{3.12}$$

and write $\delta^v(t_n)$ for the sum of the terms in the braces of eq. (3.4d), i.e.,

$$\delta^v(t_n) = v(t_{n+1}) - \left(v(t_n) + h\sum_{i=1}^{s} b_i v'(t_{ni})\right). \tag{3.13}$$

It is easy to tell that the function $\delta^v(t)$ can be regarded as the local error of a RK method for ODEs and consequently equals to $O(h^{\kappa+1})$. With the above notation defined, the convergence of the IRK-CP scheme (3.3) can be stated as follows:

Theorem 1. *Consider an index-1 differential-algebraic system of eq.* (2.1) *and suppose that its initial conditions satisfy eq.* (2.2)*, and that* $(MG)^{-1}$ *exists and is bounded. Given an IRK scheme that approximates the solution of the system with eq.* (3.3) *for* u *and eq.* (2.7) *or* $p_{n+1} = P_{ns}$ *for* p*. Suppose that its coefficients are of type I or type II, define a standard Runge-Kutta method of order* κ *for ODEs and satisfy* $C(\zeta)$ *with* $\kappa \geq \max(\zeta, 1) + 1$*. Also, suppose that the perturbations* δ_{n+1}*,* δ_{ni}*, and* θ_{ni} *in eq.* (3.3) *are of* $O(h^\kappa)$*, then global errors for the* u*- and* p*-component given by the method satisfy, for* $h \leq h_0$ *and* $t_n = nh \leq Const$*, the estimates*

$$u_n - u(t_n) = O(h^\kappa), \quad p_n - p(t_n) = \begin{cases} O(h^{\zeta+1}) & \textit{for type I coefficients with } |R(\infty)| < 1, \\ O(h^\zeta) & \textit{for type I coefficients with } |R(\infty)| = 1, \\ O(h^\kappa) & \textit{for stiffly accurate type I and type II coefficients.} \end{cases} \tag{3.14}$$

If we further assume the perturbations δ_{n+1}, δ_{ni}, and θ_{ni} admit asymptotic expansions of the form

$$\begin{aligned} \delta_{n+1} &= h^\kappa c_\kappa(t_n) + h^{\kappa+1} c_{\kappa+1}(t_n) + \cdots + h^N c_N(t_n) + O(h^{N+1}), \\ \delta_{ni} &= h^\kappa C_{i,\kappa}(t_n) + h^{\kappa+1} C_{i,\kappa+1}(t_n) + \cdots + h^N C_{i,N}(t_n) + O(h^{N+1}), \\ \theta_{ni} &= h^\kappa D_{i,\kappa}(t_n) + h^{\kappa+1} D_{i,\kappa+1}(t_n) + \cdots + h^N D_{i,N}(t_n) + O(h^{N+1}), \end{aligned} \tag{3.15}$$

the order of $p_n - p(t_n)$, for the type I coefficients with $R(\infty) = -1$ satisfied in addition, can be sharpened with $O(h^{\zeta+1})$.

Remark 1. The numerical solution (u_n, p_n) given by the scheme (3.3) enforces the algebraic constraint of eq. (1.5b) exactly when the constraints of eq. (3.4) are satisfied at each numerical step.

The proof of Theorem 1 is available in Appendix E. Its statement indicates that, when the introduced perturbations are sufficiently small, the IRK-CP scheme can attain the same orders of convergence as those of the IRK-DAE1 scheme while satisfying the algebraic constraint (i.e., the continuity equation) exactly, at least for RK coefficients of type II and type I with $-1 < R(\infty) \le 1$. To make this convergence property also hold for the type I coefficients with $R(\infty) = -1$, the implicit equations in the IRK-CP scheme should be solved with sufficient accuracy (i.e., δ^u_{ni} and θ^u_{ni} approach to zero) such that the additional assumption of eq. (3.15) can be satisfied given properly determined perturbations.

3.2. Determination of the prescribed perturbations in the IRK-CP scheme

We now discuss the determination of the four intentionally prescribed perturbations, δ^u_{n+1}, δ^v_{n+1}, δ^v_{ni}, and θ^v_{ni}) in the IRK-CP scheme (3.3). The other two perturbations, representing solution residual errors, are computed from eqs. (3.3b) and (3.3c) after solving the corresponding intermediate stage equations. First, δ^u_{n+1} can be directly computed from eqs. (3.4a) and (3.4b) for non-stiffly accurate methods, in which the values of δ^u_{ni} will be available once the numerical solution of (U_{ni}, P_{ni}) is obtained. For stiffly accurate methods, there is no need to compute δ^u_{n+1} since $u_{n+1} = U_{ns}$. It is also useless to compute δ^v_{n+1} considering its value is not required by the solution of RK intermediate stages. We only need to set $v_{n+1} = v(t_{n+1})$ at the end of each time step. Concurrently, δ^v_{ni} are explicitly given by θ^v_{ni} (whose values will be discussed below) according to eq. (3.4c).

For the last perturbation θ^v_{ni}, we first consider the situation in which the explicit expression of the boundary velocity $v(t)$ is given, so is its time-derivative $v'(t)$. It is evident to tell eq. (3.4) does not uniquely define θ^v_{ni}, so the specification of their values can be quite arbitrary. For IRK methods of type I, it can be assumed that θ^v_{ni} are linearly dependent on c_i. However, for IRK methods of type II, there exists a hidden constraint concerning θ^v_{ni} which requires extra attention. Since we have $U_{ns} = u_{n+1} = U_{n+1,1}$ (where $U_{n+1,1}$ denotes the value of u at the first intermediate stage when the scheme marches from t_{n+1} to t_{n+2}), it is necessary to ensure $\theta^v_{ns} = \theta^v_{n+1,1}$ for the consistency of the IRK-CP scheme. A simple remedy would be to set the θ^v_{ni} at the first and last intermediate stages to zero. Assuming the non-zero θ^v_{ni} are still linearly dependent on c_i, we have

$$\theta^v_{ni} = \frac{\bar{c}_i}{h b^T \bar{c}} \delta^v(t_n), \quad \bar{c} = \begin{cases} c & \text{for type I coefficients,} \\ (0, c_2, \cdots, c_{s-1}, 0)^T & \text{for type II coefficients with } s \ge 3. \end{cases} \tag{3.16}$$

Considering $\delta^v(t)$ is of order $O(h^{\kappa+1})$, the requirement of Theorem 1 on the order of the perturbations is satisfied. Replacing the $v(t_{n+1})$ and $v(t_{ni})$ in eq. (3.13) with their Taylor expansions, $\delta^v(t_n)$ can then be shown to admit an asymptotic expansion of the form

$$\delta^v(t_n) = h^{\kappa+1} \left(\frac{1}{(\kappa+1)!} - b^T c^\kappa \right) v^{(\kappa+1)}(t_n) + \cdots + h^{N+1} \left(\frac{1}{(N+1)!} - b^T c^N \right) v^{(N+1)}(t_n) + O(h^{N+2}), \tag{3.17}$$

where the superscript of $v(t_n)$ denotes its differential order, $c^\kappa = (c_1^\kappa, \cdots, c_s^\kappa)^T$ and similarly for c^N. From eq. (3.17), it is evident that the additional assumption of eq. (3.15) in Theorem 1 can be satisfied when the residual errors δ^u_{ni} and θ^u_{ni} are sufficiently small.

The use of the IRK-CP scheme (3.3), with θ^v_{ni} determined from eq. (3.16), requires the explicit expression of $v'(t)$. If this requirement cannot be met in practice, an alternative approach is to approximate $v'(t_{ni})$ using the values of $v(t)$ at discrete-time. In order to do so, we consider a Lagrange interpolation polynomial $\bar{v}(t)$ constructed by a total of $\kappa + 1$ distinct interpolation nodes selected from $[t_n, t_n + h]$, i.e.,

$$\bar{v}(t) = \sum_{i=1}^{\kappa+1} v(t_n + \phi_i h) l_i(t), \tag{3.18}$$

where $l_i(t)$ are Lagrange interpolation polynomials of the form

$$l_i(t) = \prod_{j \in B_i} \frac{t - (t_n + \phi_j h)}{(\phi_i - \phi_j) h}, \tag{3.19}$$

with $B_i = \{j \mid j \neq i, j \in \{1, 2, \cdots, \kappa + 1\}\}$ and $\phi_i \in [0, 1]$. In the following context, we write ϕ for the vector of $(\phi_1, \cdots, \phi_{\kappa+1})$ for simplicity. Once the Lagrange approximation $\bar{v}(t)$ is obtained, we can compute θ^v_{ni} from

$$\theta^v_{ni} = \bar{v}'(t_{ni}) - v'(t_{ni}) + \frac{c_i}{hb^T\bar{c}}\bar{\delta}^v(t_n), \tag{3.20}$$

where $\bar{c}$ is identical to that of eq. (3.16) and $\bar{\delta}^v(t_n)$ is defined as

$$\bar{\delta}^v(t_n) = v(t_{n+1}) - \left(v(t_n) + h\sum_{i=1}^{s} b_i\bar{v}'(t_{ni})\right). \tag{3.21}$$

Considering the Lagrange polynomial $\bar{v}(t)$ is of order κ in approximating $v(t)$ (yielding $||\bar{v}'(t) - v'(t)|| = O(h^\kappa)$), one can easily verify that θ^v_{ni} given by eq. (3.20) and the resulting δ^v_{ni} also meet the requirement of Theorem 1 on their orders of magnitudes. It can be easily verified that the additional assumption of eq. (3.15) in Theorem 1 can also be met in this situation. Inserting eq. (3.20) into eqs. (3.3b) and (3.3c), it becomes evident that the resulting IRK-CP scheme only demands for the boundary velocities at the time-instants of t_n and $t_n + \phi_i h$ (for $i = 1, \cdots, \kappa + 1$). By selecting the interpolation nodes ϕ_i according to the available discrete-time values of $v(t)$, the IRK-CP scheme no longer requires $v(t)$ as an explicit function of time. Using eq. (3.16) or eq. (3.20), the V_{ni} given by the second relation of eq. (3.3b) can be rewritten as

$$V_{ni} = \begin{cases} v(t_n) + h\sum_{j=1}^{s} a_{ij}\left(v'(t_{nj}) + c_j\delta^v(t_n)/(hb^T\bar{c})\right) & \text{if } v(t) \text{ is explicitly available,} \qquad \text{(a)} \\ v(t_n) + h\sum_{j=1}^{s} a_{ij}\left(\bar{v}'(t_{nj}) + c_j\bar{\delta}^v(t_n)/(hb^T\bar{c})\right) & \text{if } v(t) \text{ is not explicitly available.} \quad \text{(b)} \end{cases} \tag{3.22}$$

3.3. Solution of the intermediate stage equations in the IRK-CP scheme

At the end of this section, we have a brief discussion on the solution of the RK intermediate stage velocity and pressure (U_{ni}, P_{ni}) in the IRK-CP scheme. By setting the residual error δ^u_{ni} in the first relation of eq. (3.3b) to zero and using eq. (3.2) for U'_{nj}, the fully-discrete momentum equation at the intermediate stages of the scheme writes, for $i = 1, \cdots, s$,

$$U_{ni} = u_n + h\sum_{j=1}^{s} a_{ij}\left[-C(U_{nj})U_{nj} + \nu DU_{nj} - GP_{nj} + r(t_{nj}, V_{nj})\right], \tag{3.23}$$

where V_{nj} takes eq. (3.22). The IRK-CP scheme requires that, for non-stiffly accurate IRK methods, the initial and all previous discrete-time velocity fields be divergence-free to ensure that the divergence-free constraint is enforced at the new time step, i.e., $g(t_{n+1}, u_{n+1}) = 0$. However, due to round-off or residual errors in the solution, this condition may not always be exactly satisfied in practice, potentially leading to the accumulation of errors in the continuity equation over time. A straightforward yet effective approach to address this issue is to solve, instead of eq. (3.8), the intermediate-stage continuity equations, expressed for $i = 1, \cdots, s$ as

$$MU_{ni} + NV_{ni} + \epsilon_i = 0, \tag{3.24}$$

where

$$\epsilon_i = \begin{cases} 0 & \text{if RK coefficients are stiffly accurate,} \\ \frac{g(t_n, u_n)}{\sum_{j=1}^{s} b_i\omega_{ij}} & \text{if RK coefficients are non-stiffly accurate.} \end{cases} \tag{3.25}$$

From eqs. (3.24) and (3.25), it can be demonstrated that the u_{n+1} computed from eq. (3.9) enforces $g(t_{n+1}, u_{n+1}) = 0$ even when $g(t_n, u_n) \neq 0$. The term ϵ_i in eq. (3.24) can be interpreted as a round-off error, whose magnitude is significantly smaller than that of the time discretization error. Therefore, its impact on the convergence of the IRK methods is expected to be negligible. From eqs. (3.23) and (3.24), it is evident to tell the IRK-CP scheme, compared to the conventional IRK-DAE2 scheme, essentially impose perturbed values for the unsteady inflow velocity at RK intermediate stages, revising the $v(t_{ni})$ in the original fully-discrete N-S equations to the V_{ni} of eq. (3.22). This makes the implementation of the scheme easy to achieve on the existing CFD code.

Identical to the IRK-DAE1 and IRK-DAE2 schemes, there are a total of $(m + n) \times s$ implicit equations to be solved within one step for the IRK-CP scheme. Apparently, eqs. (3.23) and (3.24) can be solved with a general non-linear equation solver, the details of which will not be discussed here. This study is more concerned with the special case in which the coefficient matrix $[a_{ij}]$ is lower triangular, in other words, the IRK method is diagonally implicit. In this case, eq. (3.23) reduces to

$$U_{ni} = u_{ni} + ha_{ii}\left(-C(U_{ni})U_{ni} + \nu DU_{ni} - GP_{ni} + r_{ni}\right), \tag{3.26}$$

where u_{ni} and r_{ni} are explicit terms given by

$$u_{ni} = u_n + h\sum_{j=1}^{i-1} a_{ij}U'_{nj}, \quad r_{ni} = r(t_{ni}, V_{ni}). \tag{3.27}$$

Introducing the notation A_{ni} and H_{ni} defined as

$$A_{ni} = \frac{1}{h} + a_{ii}\text{diag}\left(-\nu D + C(U_{ni})\right), \quad H_{ni} = a_{ii}\text{offdiag}\left(\nu D - C(U_{ni})\right), \tag{3.28}$$

where diag() and offdiag() are two operators returning the diagonal and off-diagonal elements of a matrix, respectively, eq. (3.26) can be rewritten as

$$\left(A_{ni} - H_{ni}\right) U_{ni} + a_{ii} G P_{ni} = \frac{u_{ni}}{h} + a_{ii} r_{ni}. \tag{3.29}$$

Utilizing eq. (3.29) and the non-singularity of A_{ni}, eq. (3.24) can be replaced with an equality that is expressed as

$$a_{ii} M A_{ni}^{-1} G P_{ni} = M A_{ni}^{-1}\left(\frac{u_{ni}}{h} + a_{ii} r_{ni} + H_{ni} U_{ni}\right) + N V_{ni} + \epsilon_i. \tag{3.30}$$

When investigating the solution of eqs. (3.29) and (3.30), it is advantageous to group them into a matrix equation of the form

$$\begin{pmatrix} A_{ni} - H_{ni} & a_{ii} G \\ -M A_{ni}^{-1} H_{ni} & a_{ii} M A_{ni}^{-1} G \end{pmatrix} \begin{pmatrix} U_{ni} \\ P_{ni} \end{pmatrix} = \frac{1}{h} \begin{pmatrix} u_{ni} + h a_{ii} r_{ni} \\ M A_{ni}^{-1}\left(u_{ni} + h a_{ii} r_{ni}\right) + h N V_{ni} + h\epsilon_i \end{pmatrix}. \tag{3.31}$$

Thus, the coupling between U_{ni} and P_{ni} can be clearly noted. The system of eq. (3.31) can be linearized by computing A_{ni} and H_{ni} using the U_{ni} acquired from previous Picard iterations. The linearized system can then be solved with the PISO [47] algorithm which decouples the velocity-pressure through an iteration formula of the form

$$\begin{pmatrix} A_{ni} & a_{ii} G \\ 0 & a_{ii} M A_{ni}^{-1} G \end{pmatrix} \begin{pmatrix} U_{ni} \\ P_{ni} \end{pmatrix} = \begin{pmatrix} H_{ni} & 0 \\ M A_{ni}^{-1} H_{ni} & 0 \end{pmatrix} \begin{pmatrix} U_{ni} \\ P_{ni} \end{pmatrix} + \frac{1}{h} \begin{pmatrix} u_{ni} + h a_{ii} r_{ni} \\ M A_{ni}^{-1}\left(u_{ni} + h a_{ii} r_{ni}\right) + h N V_{ni} + h\epsilon_i \end{pmatrix}. \tag{3.32}$$

In eq. (3.31), the (U_{ni}, P_{ni}) on the left and the right-hand sides of the equation denote the solution from the current and the previous velocity-pressure decoupling iterations, respectively. As the block matrix on the left-hand side of eq. (3.32) is upper triangular, the unknown velocity and pressure are now decoupled and can be solved successively. After the iterative solution of eq. (3.31), the obtained U_{ni} can be used to update A_{ni} and H_{ni} in the next Picard iteration. In practice, either the Picard or the PISO iteration will be terminated before the exact solutions are reached. As suggested by Theorem 1, if the residual errors are constrained within certain magnitudes, the convergence for the IRK-CP scheme will not be affected.

After the solution of (U_{ni}, P_{ni}), the velocity and pressure at the new time-step are given by (U_{ns}, P_{ns}) for stiffly accurate RK coefficients, and by the first relation of eq. (3.3a) and eq. (2.7) for non-stiffly accurate RK coefficients. An algorithm, applying the IRK-CP scheme with diagonally implicit RK coefficients to the solution of the semi-discrete system (1.2) from t_n to t_{n+1}, is presented in Algorithm 1. In the algorithm, the initial value for (U_{ni}, P_{ni}), required to trigger the iterative solution procedure of intermediate-stage N-S equations, takes (u_n, p_n) or $(U_{n,i-1}, P_{n,i-1})$ for simplicity. In practice, an estimation being closer to their exact solution can be employed to accelerate the convergence of the solution.

4. Linear and nonlinear stability of RK methods

In addition to the convergence order, stability is a key factor in selecting time discretization methods for semi-discrete Navier-Stokes systems. Methods with better stability properties allow larger time steps, which improves efficiency. While implicit methods generally provide superior stability compared to explicit methods, they come with higher computational costs. This raises the question: which methods, implicit or explicit, are more efficient in practice? For semi-discrete Navier-Stokes systems containing both linear and nonlinear terms, it's difficult to provide a single estimate of time-step and grid sizes for comparing the efficiency of implicit and explicit methods. Therefore, analyzing stability through both linear and nonlinear estimates can be useful.

A common estimate of linear stability is the stability function $R(z)$, which represents the ratio between successive numerical solutions

$$y_{n+1} = R(z) y_n, \tag{4.1}$$

from a numerical method applied to the Dahlquist test equation

$$y' = \lambda y, \tag{4.2}$$

where $z = \lambda h \in \mathbb{C}$. The stability region is defined by the values of z for which $|R(z)| \le 1$. A method is $A(\alpha)$-stable if $|R(z)| \le 1$ for $|\arg(-z)| \le \alpha$, and A-stable if $\alpha = 90°$. If, in addition, $R(\infty) = 0$, the method is $L(\alpha)$-stable or L-stable if $\alpha = 90°$. IRK methods, particularly A-stable or L-stable ones, are preferred for their unrestricted stability when $\text{Re}(\lambda) < 0$ and $h > 0$. The property of L-stability, specifically $R(\infty) = 0$, is also favorable for the convergence of IRK methods for the algebraic component in DAEs, such as pressure in the semi-discrete Navier-Stokes system. In contrast, ERK methods require an upper bound on h when $\text{Re}(\lambda) < 0$ to maintain $|R(z)| \le 1$.

Nonlinear stability is commonly evaluated using the strong stability preserving (SSP) property. Consider a semi-discrete ODE

$$y' = f(y), \tag{4.3}$$

and assume that the forward Euler method applied to this ODE satisfies

$$\left\| y_n + h f(y_n) \right\| \le \left\| y_n \right\| \quad \text{for } 0 \le h \le h_{\text{FE}}, \tag{4.4}$$

Algorithm 1 An algorithm of the IRK-CP scheme with diagonally implicit RK coefficients applied to solving eq. (1.2).

```
if $v(t)$ is explicitly given then
    $\delta^v(t_n) \leftarrow v(t_{n+1}) - \left[v(t_n) + h\sum_{i=1}^s b_i v'(t_{ni})\right]$
    $V_{ni} \leftarrow v(t_n) + h\sum_{j=1}^s a_{ij}\left[v'(t_{nj}) + c_j\delta^v(t_n)/(hb^T\bar{c})\right],\ i = 1(1)s$
else
    $\bar{v}(t) \leftarrow \sum_{i=1}^{\kappa+1} v(t_n + \phi_i h) l_i(t)$
    $\bar{\delta}^v(t_n) \leftarrow v(t_{n+1}) - \left[v(t_n) + h\sum_{i=1}^s b_i \bar{v}'(t_{ni})\right]$
    $V_{ni} \leftarrow v(t_n) + h\sum_{j=1}^s a_{ij}\left[\bar{v}'(t_{nj}) + c_j\bar{\delta}^v(t_n)/(hb^T\bar{c})\right],\ i = 1(1)s$
end if
for $i \leftarrow 1$ to $s$ do
    if $i = 1$ then                                                    ▷ set the initial value of $(U_{ni}, P_{ni})$
        $U_{ni} \leftarrow u_n,\ P_{ni} \leftarrow p_n$
    else
        $U_{ni} \leftarrow U_{n,i-1},\ P_{ni} \leftarrow P_{n,i-1}$
    end if
    $u_{ni} \leftarrow u_n + h\sum_{j=1}^{i-1} a_{ij} U'_{nj},\ r_{ni} \leftarrow r(t_{ni}, V_{ni})$
    if stiffly accurate then
        $\epsilon_i \leftarrow 0$
    else
        $\epsilon_i \leftarrow g(t_n, u_n)/\sum_{j=1}^s b_i\omega_{ij}$
    end if
    for Picard iteration loop $\leftarrow 1$ to $N_{\text{Picard}}$ do
        $A_{ni} \leftarrow 1/h + a_{ii}\text{diag}\left(-\nu D + C(U_{ni})\right),\ H_{ni} = a_{ii}\text{offdiag}\left(\nu D - C(U_{ni})\right)$    ▷ Picard linearization
        $U_{ni} \leftarrow \left(A_{ni} - H_{ni}\right)U_{ni} + a_{ii}GP_{ni} = u_{ni}/h + a_{ii}r_{ni}$    ▷ solve the momentum equation regarding $U_{ni}$
        for PISO iteration loop $\leftarrow 1$ to $N_{\text{PISO}}$ do
            $P_{ni} \leftarrow a_{ii}MA_{ni}^{-1}GP_{ni} = MA_{ni}^{-1}\left(u_{ni}/h + a_{ii}r_{ni} + H_{ni}U_{ni}\right) + NV_{ni} + \epsilon_i$    ▷ solve the Poisson equation regarding $P_{ni}$
            $U_{ni} \leftarrow A_{ni}^{-1}\left(u_{ni}/h + a_{ii}r_{ni} + H_{ni}U_{ni} - a_{ii}GP_{ni}\right)$
        end for
        if Picard iteration loop $=$ $N_{\text{Picard}}$ then
            $U'_{ni} \leftarrow -C(U_{ni})U_{ni} + \nu DU_{ni} - GP_{ni} + r_{ni}$
            $\delta^u_{ni} \leftarrow u_n + h\sum_{j=1}^i a_{ij}U'_{nj} - U_{ni}$
        end if
    end for
end for
if stiffly accurate then
    $u_{n+1} \leftarrow U_{ns}$
    $p_{n+1} \leftarrow P_{ns}$
else
    $h\theta^u_{ni} \leftarrow \sum_j^s \omega_{ij}\delta^u_{nj},\ i = 1(1)s$
    $h\delta^u_{n+1} \leftarrow h\sum_{i=1}^s b_i\theta^u_{ni}$
    $u_{n+1} \leftarrow u_n + h\sum_{i=1}^s b_i U'_{ni} + h\delta^u_{n+1}$
    $p_{n+1} \leftarrow p_n + \sum_{i,j=1}^s b_i\omega_{ij}(P_{nj} - p_n)$
end if
```

where $\|\cdot\|$ is a convex functional. A Runge-Kutta method is SSP with the SSP coefficient c_{ssp} if its solution satisfies

$$\left\|y_{n+1}\right\| \le \left\|y_n\right\|, \tag{4.5}$$

whenever the time-step size h meets the condition

$$h \le c_{\text{ssp}} h_{\text{FE}} \quad \text{for some } c_{\text{ssp}} > 0. \tag{4.6}$$

To compare the efficiency of SSP-RK methods, the effective SSP coefficient c_{eff} [48] is defined as

$$c_{\text{eff}} = \frac{c_{\text{ssp}}}{s}, \tag{4.7}$$

where s is the number of stages. According to literature, optimal explicit SSP-RK methods have $c_{\text{eff}} \le 1$ [49], while optimal implicit methods have $c_{\text{eff}} \le 2$ [50]. This suggests that, from a nonlinear stability perspective, implicit methods may not be as efficient as explicit methods due to their higher computational cost. Among IRK methods, a class of 2nd-order SDIRK methods with $c_{\text{eff}} = 2$ is optimal [50], and the 1-stage method in this class is the implicit midpoint rule (or the 1-stage Gauss method with RK coefficients given in Table 5.1). The s-stage method of this class can be interpreted as s-successive applications of the implicit midpoint rule. It has $R(\infty) = -1$ if s is odd and $R(\infty) = 1$ if s is even. Note that the property of $R(\infty) = -1$ is more favored for the algebraic component convergence of IRK methods in DAEs, compared to $R(\infty) = 1$. As a result, the implicit midpoint rule is a good choice for DAEs, balancing stability and computational efficiency. Meanwhile, the optimal class of 3rd-order IRK methods has $|R(\infty)| > 1$, which makes it not suitable for DAEs. More details concerning optimal higher-order SSP IRK methods can be found in the study of [50].

It is important to note that these stability estimates were developed for ODEs. The maximum allowable time step for a particular RK method based on these estimates may differ when applied to DAEs. Nonetheless, they offer valuable guidance for choosing RK methods in practice. From a linear stability perspective, IRK methods are generally more efficient than ERK methods. However,

Table 5.1
A 1-stage Gauss method with A-stability and $c_{\text{ssp}} = 2$.

$1/2$	$1/2$
	1

Table 5.2
A 2-stage Gauss method with A-stability and $c_{\text{ssp}} = 0$.

$1/2 - \sqrt{3}/6$	$1/4$	$1/4 - \sqrt{3}/6$
$1/2 + \sqrt{3}/6$	$1/4 + \sqrt{3}/6$	$1/4$
	$1/2$	$1/2$

Table 5.3
A 2-stage Radau IA method with L-stability and $c_{\text{ssp}} = 0$.

0	$1/4$	$-1/4$
$2/3$	$1/4$	$5/12$
	$1/4$	$3/4$

Table 5.4
A 2-stage Radau IIA method with L-stability and $c_{\text{ssp}} = 0$.

$1/3$	$5/12$	$-1/12$
1	$3/4$	$1/4$
	$3/4$	$1/4$

the fact that optimal implicit SSP-RK methods do not have significantly larger c_{eff} values than explicit ones calls into question the advantage of implicit schemes. To the authors' knowledge, there has been no comparative efficiency study of IRK and ERK methods in terms of nonlinear stability for DAEs. This is a topic that deserves further investigation. The first systematic comparison of IRK, ERK, and IMEX-RK methods for ODEs was done recently in [48], where explicit SSP-RK methods were found to be the most efficient. However, it was also argued that the test cases used may not have been stiff enough to demonstrate the competitiveness of implicit SSP-RK methods. In this study, a preliminary efficiency analysis of implicit methods and half-explicit methods for semi-discrete incompressible Navier-Stokes systems is presented in the following numerical experiments.

5. Numerical experiments

5.1. A simplified test problem

In order to verify the results pertaining to the order of convergence for the IRK-DAE1, IRK-DAE2, and SRK-DAE2 schemes, as summarized in Table 2.2, along with the convergence of the IRK-CP scheme (3.3), we examine a test problem, fulfilling hypotheses H1 to H3, which writes

$$u_1' = u_1 u_2^3 p, \quad u_2' = -\frac{1}{2} u_2^4 p, \quad u_1 = u_2 + v(t), \tag{5.1}$$

with $v(t) = e^t - e^{-t/2}$. Given consistent initial values $u_0 = (1, 1)$ and $p_0 = 1$, the exact solution of this problem can be written as $u_1(t) = e^t$, $u_2(t) = e^{-t/2}$, and $p(t) = e^{3t/2}$. For the application of the IRK-CP scheme in this numerical experiment, eq. (3.22a) is selected for the determination of V_{ni}, in which $v(t)$ and $v'(t)$ are assumed to be explicitly given.

Thirteen methods, selected from [13,32,37], encompassing a 1-stage and a 2-stage Gauss method, a 2-stage Radau IA method, a 2-stage Radau IIA method, a 3-stage Lobatto IIIA method, a 3-stage Lobatto IIIC method, a 2-stage, a 3-stage, a 4-stage, and a 5-stage SDIRK method, a 3-stage, a 4-stage, and a 6-stage ESDIRK method, are examined. All considered IRK methods are A-stable [51], L-stable [52] or $L(\alpha)$-stable [53], and their RK and SSP coefficients are provided in Tables 5.1 to 5.13, respectively. A Python code has been developed to perform the numerical experiment under consideration. Given that numerical truncation errors may constitute a significant portion of the total computation errors for the problems (when the errors resulting from the numerical methods are sufficiently small), the convergence orders of the numerical solutions could be affected. To mitigate the influence of numerical truncation errors, the mpmath package is employed, utilizing a higher-precision floating-point numeric type.

The global errors for the u- and p-components at $t = 1$ resulting from the IRK-DAE1, IRK-DAE2, SRK-DAE2, and IRK-CP schemes with varying step-sizes h are illustrated in Figs. 1 and 2, respectively. Logarithmic scales are employed in the plots, causing an error

Table 5.5
A 3-stage Lobatto IIIA method with L-stability and $c_{\text{ssp}} = 0$.

0	0	0	0
1/2	5/24	1/3	−1/24
1	1/6	2/3	1/6
	1/6	2/3	1/6

Table 5.6
A 3-stage Lobatto IIIC method with L-stability and $c_{\text{ssp}} = 0$.

0	1/6	−1/3	1/6
1/2	1/6	5/12	−1/12
1	1/6	2/3	1/6
	1/6	2/3	1/6

Table 5.7
A 2-stage SDIRK method with L-stability and $c_{\text{ssp}} = \sqrt{2} + 1$.

$1 - \sqrt{2}/2$	$1 - \sqrt{2}/2$	
1	$\sqrt{2}/2$	$1 - \sqrt{2}/2$
	$\sqrt{2}/2$	$1 - \sqrt{2}/2$

Table 5.8
A 3-stage SDIRK method with L-stability and $c_{\text{ssp}} = 0$: $c_2 = (1 + \gamma)/2$, $b_2 = (6\gamma^2 - 20\gamma + 5)/4$ and γ is the root of $6\gamma^3 - 18\gamma^2 + 9\gamma - 1 = 0$ lying in $(1/6, 1/2)$.

γ	γ		
c_2	$c_2 - \gamma$	γ	
1	$1 - \gamma - b_2$	b_2	γ
	$1 - \gamma - b_2$	b_2	γ

Table 5.9
A 4-stage SDIRK method with L-stability and $c_{\text{ssp}} = 0$.

1/4	1/4			
11/28	1/7	1/4		
1/3	61/144	−49/144	1/4	
1	0	0	3/4	1/4
	0	0	3/4	1/4

Table 5.10
A 5-stage SDIRK method with L-stability and $c_{\text{ssp}} = 0$.

1/4	1/4				
3/4	1/2	1/4			
11/20	17/50	−1/25	1/4		
1/2	371/1360	−137/2720	15/544	1/4	
1	25/24	−49/48	125/16	−85/12	1/4
	25/24	−49/48	125/16	−85/12	1/4

Table 5.11
A 3-stage ESDIRK method with L-stability and $c_{\text{ssp}} = \sqrt{2} + 1$.

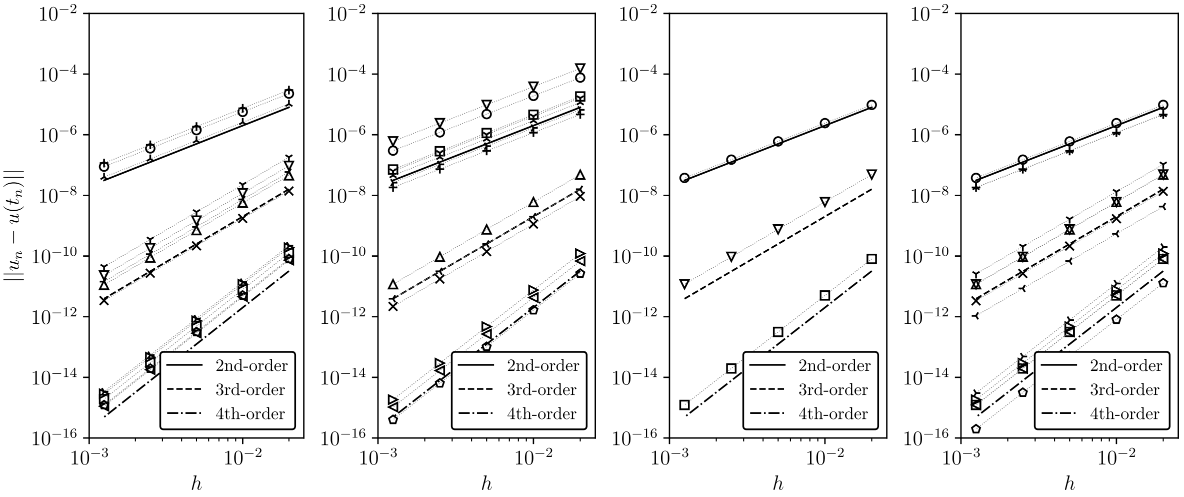
0	0		
$2 - \sqrt{2}$	$1 - \sqrt{2}/2$	$1 - \sqrt{2}/2$	
1	$\sqrt{2}/4$	$\sqrt{2}/4$	$1 - \sqrt{2}/2$
	$\sqrt{2}/4$	$\sqrt{2}/4$	$1 - \sqrt{2}/2$

Table 5.12
A 4-stage ESDIRK method with $L(75.6°)$-stability and $c_{\text{ssp}} = 1.8868$: γ is the root of $6\gamma^3 - 18\gamma^2 + 9\gamma - 1 = 0$ lying in $(3/20, 4/25)$, $b_3 = \left(\sqrt{2} - 1\right)\left(6\gamma^2 - 6\gamma + 1\right) / \left(6\gamma^2\right)$.

0	0			
2γ	γ	γ		
$\left(2+\sqrt{2}\right)\gamma$	$\left(c_3-\gamma\right)/2$	$\left(c_3-\gamma\right)/2$	γ	
1	$\left(1-b_3-\gamma\right)/2$	$\left(1-b_3-\gamma\right)/2$	b_3	γ
	$\left(1-b_3-\gamma\right)/2$	$\left(1-b_3-\gamma\right)/2$	b_3	γ

Table 5.13
A 6-stage ESDIRK method with $L(89.95°)$-stability and $c_{\text{ssp}} = 0$.

0	0					
$1/3$	$1/6$	$1/6$				
$8/15$	$31/150$	$4/25$	$1/6$			
$1/5$	$1342/13200$	$-191/1650$	$25/528$	$1/6$		
$1/2$	$65/288$	$103/144$	$-65/288$	$-55/144$	$1/6$	
1	$1/6$	0	0	0	$2/3$	$1/6$
	$1/6$	0	0	0	$2/3$	$1/6$



(a) IRK-DAE1 scheme (b) IRK-DAE2 scheme (c) SRK-DAE2 scheme (d) IRK-CP scheme

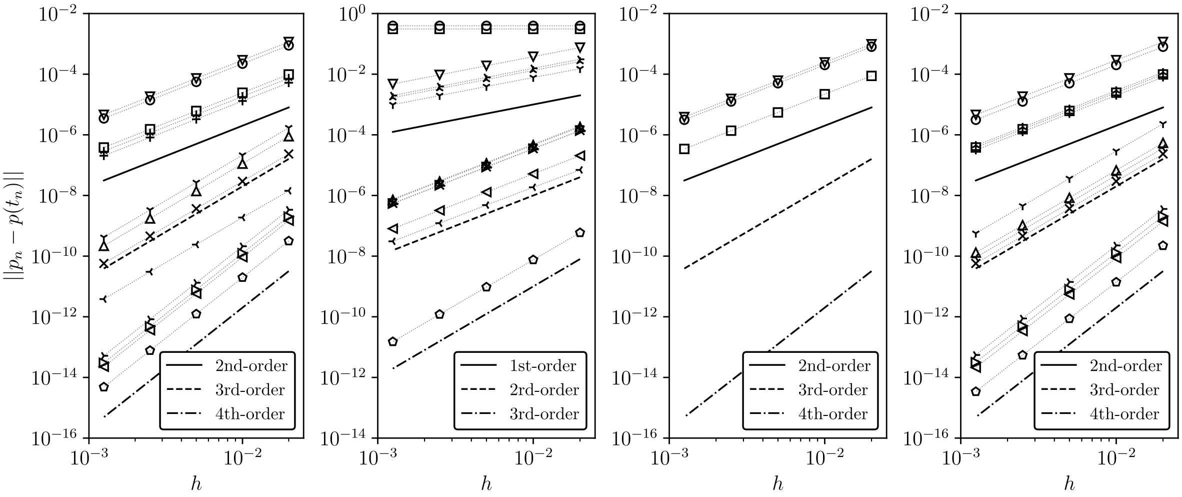
Fig. 1. Global errors for the u-component at $t = 1$ for various RK methods implemented with different schemes: 1-stage Gauss ○; 2-stage Gauss □; 2-stage Radau IA ▽; 2-stage Radau IIA △; 3-stage Lobatto IIIA ◁; 3-stage Lobatto IIIC ▷; 2-stage SDIRK ⅄; 3-stage SDIRK Y; 4-stage SDIRK ≺; 5-stage SDIRK ≻; 3-stage ESDIRK +; 4-stage ESDIRK ×; 6-stage ESDIRK ⬠.

curve to appear as a straight line with a slope of k if the leading term in the error is of order $O(h^k)$. All global error curves depicted in these plots are consistent with the order results summarized in Table 2.2 or predicted by Theorem 1. It can be observed that the IRK-DAE1 and IRK-CP schemes achieve identical order results for all selected RK methods, irrespective of whether the u- or p-component is considered.

Table 5.14 presents the residual error of the algebraic constraint, that is, the third relation of eq. (5.1), at $t = 1$ with a constant step-size of $h = 0.02$, for the investigated IRK methods, which are respectively implemented with the IRK-DAE1, IRK-DAE2, SRK-DAE2, and IRK-CP schemes. Both the SRK-DAE2 and IRK-CP schemes ensure the exact satisfaction of the algebraic constraint, whereas the IRK-DAE2 scheme only fulfills this requirement in the case of stiffly-accurate RK coefficients. In contrast, the IRK-DAE1 scheme is unable to precisely satisfy the algebraic constraint for all types of RK coefficients.

5.2. A two-dimensional backward-facing step flow with unsteady inflow

In the second numerical experiment, the temporal convergence of the IRK-CP scheme, employing diagonally implicit RK coefficients, is investigated by simulating a two-dimensional backward-facing step (BFS) flow with unsteady inflow. A preliminary efficiency analysis of IRK and HERK methods, with the stiffness of the semi-discrete Navier-Stokes system taken into account, is also conducted. Different to the first experiment, eq. (3.22b) is employed for the determination of V_{ni} in the scheme, with the boundary velocity $v(t)$ assumed to be available only at uniform time instants. The 1-stage Gauss method, 2-stage and 3-stage SDIRK methods, as well as the 3-stage and 4-stage ESDIRK methods considered in the preceding experiment, are selected for examination. For half-explicit



(a) IRK-DAE1 scheme (b) IRK-DAE2 scheme (c) SRK-DAE2 scheme (d) IRK-CP scheme

Fig. 2. Global errors for the p-component at $t = 1$ for various RK methods implemented with different schemes: 1-stage Gauss ○; 2-stage Gauss □; 2-stage Radau IA ▽; 2-stage Radau IIA △; 3-stage Lobatto IIIA ◁; 3-stage Lobatto IIIC ▷; 2-stage SDIRK ⅄; 3-stage SDIRK Y; 4-stage SDIRK ≺; 5-stage SDIRK ≻; 3-stage ESDIRK +; 4-stage ESDIRK ×; 6-stage ESDIRK ⬠.

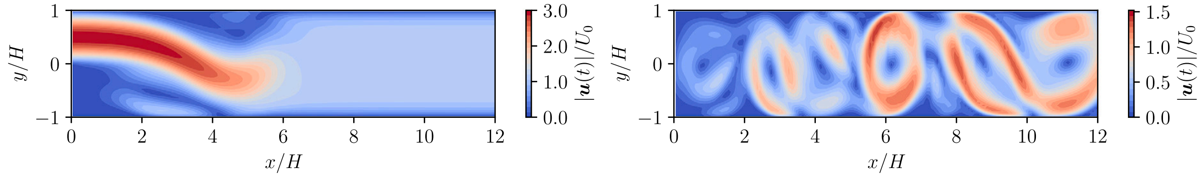
Table 5.14
The residual error of the algebraic equation at $t = 1$ with a constant step size of $h = 0.02$ for various methods implemented with the considered schemes: the notation 0* denotes an error value whose magnitude is within the solver tolerance.

method	IRK-DAE1 eq. (2.3)	IRK-DAE2 eq. (2.8)	SRK-DAE2 eq. (2.9)	IRK-CP eq. (3.3)
1-stage Gauss	3.0277×10^{-5}	9.0829×10^{-5}	0*	0*
2-stage Gauss	6.4550×10^{-11}	2.0185×10^{-5}	0*	0*
2-stage Radau IA	6.1175×10^{-8}	1.6946×10^{-4}	0*	0*
2-stage Radau IIA	6.1861×10^{-8}	0*	-	0*
3-stage Lobatto IIIA	9.6825×10^{-11}	0*	-	0*
3-stage Lobatto IIIC	9.6825×10^{-11}	0*	-	0*
2-stage SDIRK	7.4140×10^{-6}	0*	-	0*
3-stage SDIRK	1.0585×10^{-7}	0*	-	0*
4-stage SDIRK	6.1861×10^{-8}	0*	-	0*
5-stage SDIRK	3.8086×10^{-11}	0*	-	0*
3-stage ESDIRK	2.9370×10^{-5}	0*	-	0*
4-stage ESDIRK	5.3016×10^{-8}	0*	-	0*
6-stage ESDIRK	9.6825×10^{-11}	0*	-	0*

methods, a 3-stage HERK method of 3rd-order for velocity and 2nd-order for pressure, recommended in [11] as the 'best' explicit Runge-Kutta method, is selected for comparison. The 2-step and 3-step BDF methods, which attain 2nd- and 3rd-order convergence for general nonlinear index-2 DAEs respectively, are also included for comparison. It is worth noting that both the 2-stage SDIRK method and the 3-stage ESDIRK method allow for a low-stage implementation, necessitating two memory registers for velocity and a single memory register for pressure. In terms of memory usage, these two methods are comparable to the 1-step Euler method. The 1-stage Gauss method (i.e., the implicit mid-point law), which requires two memory registers for both velocity and pressure, is more memory consuming compared to them. Meanwhile, the 3-stage SDIRK method and the 4-stage ESDIRK method also enable a low-stage implementation, requiring three memory registers for velocity and one memory register for pressure, rendering it equivalent to the 2-step BDF method in terms of memory usage. Additionally, it is important to highlight that the convergence of the 3-stage ESDIRK method does not benefit from the IRK-CP scheme, as it can already achieve second-order convergence for both velocity and pressure utilizing the conventional IRK-DAE2 scheme.

In the BFS flow under consideration (see Fig. 3), fluid enters with an unsteady average velocity $U(t)$ and a parabolic magnitude profile in an upstream channel of height H_u, subsequently passing a step with a height H. The computational domain of the BFS flow extends $20H$ in the streamwise direction and is spatially discretized on two Cartesian finite volume grids (a coarser one and a finer one), respectively. The coarser grid consists of 250×40 uniformly distributed nodes in the streamwise and wall-normal directions, respectively. The finer grid employs a grid spacing which is only one tenth of the coarser grid, increasing the node number to 2500×400. Consequently, the diffusion matrix D of the semi-discrete N-S system on the finer grid is 100 times of the counterpart on the coarser grid. The fluid velocity in the upstream channel is assumed to take the form

$$v(y,t) = C_0 U(t) \left[1 - \left(1 - \frac{2y}{H_u}\right)^2\right], \quad U(t) = U_0 \left[1 - \cos\left(\frac{2\pi t}{T}\right)\right], \tag{5.2}$$



(a) $t = T/2$

(b) $t = T$

Fig. 3. Velocity magnitude contour of the considered BFS flow at the middle and the end of the simulation.

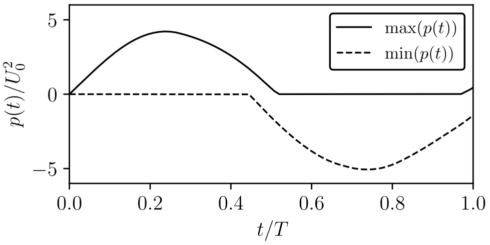


Fig. 4. Maximum and minimum values of $p(t)$ identified from the simulation of the BFS flow on the 2500×400 grid using the 4-stage ESDIRK method with $hU_0/H = 1 \times 10^{-4}$.

where $y \in [0, H_u]$ denotes the coordinate, $C_0 = 1.5$ is a normalization coefficient, U_0 is a reference velocity, and T is a time interval taken as $20H/U_0$. To accommodate the time-varying inflow velocity, the pressure field at the inlet boundary employs a time-varying fixed gradient condition based on the equality

$$\frac{\partial p}{\partial x}(y,t) = -\frac{\partial v}{\partial t}(y,t) = -C_0 \left[1 - \left(1 - \frac{2y}{H_u}\right)^2\right] \frac{\partial U}{\partial t}. \tag{5.3}$$

At the outlet boundary, the fluid velocity is assumed to have a zero-gradient in the streamwise direction, while the pressure is fixed at zero.

We simulate the considered BFS flow from $t = 0$ to T, during which the average velocity $U(t)$ increases from 0 to $2U_0$ and finally drops back to 0 again. The reference velocity U_0 is determined such that the Reynolds number, defined as UH_d/ν (where $H_d = H + H_u$ represents the height of the downstream channel), equals to 1600 at $t = T/2$. Without loss of generality, turbulence models are not considered, ensuring that the semi-discrete N-S system in this numerical experiment is identical to eq. (1.2). The initial conditions of the velocity and pressure fields are both set to zero-value fields. Since $v'(t_0) = 0$ according to eq. (5.2), the consistency condition of eq. (2.2) is satisfied regarding $u_0 = p_0 = 0$. Caused by the cosinusoidal time-varying inflow velocity and fixed-zero condition for the pressure at the outlet, the extreme value of the pressure field experiences a sinusoidal time-varying curve during the simulation, see Fig. 4.

Simulations are conducted using a custom solver for incompressible flow, implementing the IRK-CP scheme based on the open-source CFD code, OpenFOAM v2212. For finite volume solvers utilizing collocated grids (such as OpenFOAM), the computation of velocity-fluxes across cell-faces necessitates meticulous treatments to ensure that the semi-discrete N-S system, as formulated by the solvers, can be formally regarded as index-2 DAEs. Otherwise, higher-order temporal schemes may not achieve their intended orders of convergence. The cell-face velocity interpolation scheme currently employed in OpenFOAM v2212 is not applicable, and the custom solver developed in this study incorporates a new scheme that fulfills the aforementioned requirement. A brief introduction of this scheme is given in Appendix D.

The two-level iterative solution algorithm, delineated in Algorithm 1, which amalgamates the Picard iteration and the PISO algorithm, is employed for solving the fully-discrete N-S equations from IRK methods. A similar approach is used in the solver implementing BDF methods. Both IRK and BDF solvers employ two inner iteration loops for the PISO algorithm within one outer Picard iteration loop, i.e., $N_{\text{PISO}} = 2$. For solvers using the HERK method, only a Poisson equation for pressure (similar to eq. (3.30)) needs to be solved for each effective intermediate stage, where implicit equations are addressed. To standardize the notation, let s_{eff} denote the number of effective intermediate stages and N_{Poisson} represent the number of Poisson equations solved per time step. Thus, we have $N_{\text{Poisson}} = s_{\text{eff}} N_{\text{PISO}} N_{\text{Picard}}$ and $N_{\text{Poisson}} = s_{\text{eff}}$ for the employed implicit and half-explicit methods, respectively. Implicit methods have a linearized momentum equation with a strictly diagonally dominant coefficient matrix, due to a component proportional to $1/h$ on its diagonal (see eq. (3.28) for IRK methods). As a result, solving the linearized momentum equation requires significantly less computational effort compared to solving the Poisson equation for pressure. If we assume that solving the Poisson equation requires roughly the same computational effort for both implicit and half-explicit methods and neglect the effort for solving the linearized momentum equation, then N_{Poisson} serves as a reasonable measure of computational effort per time step across all considered methods. In all cases, the linear solvers provided in OpenFOAM, which are based on the Gauss-Seidel algorithm, are used to solve the linear systems for velocity and pressure, with a solution tolerance set to 10^{-11}.

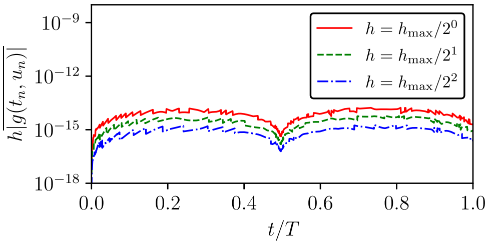


Fig. 5. Continuity equation residual errors for the 1-stage Gauss method with different time-step sizes for simulating the BFS flow on the 2500 × 400 grid.

Table 5.15
Maximum allowable dimensionless time-step sizes for the BDF, Gauss, SDIRK, ESDIRK and HERK methods applied to simulating the considered BFS flow on two different grids.

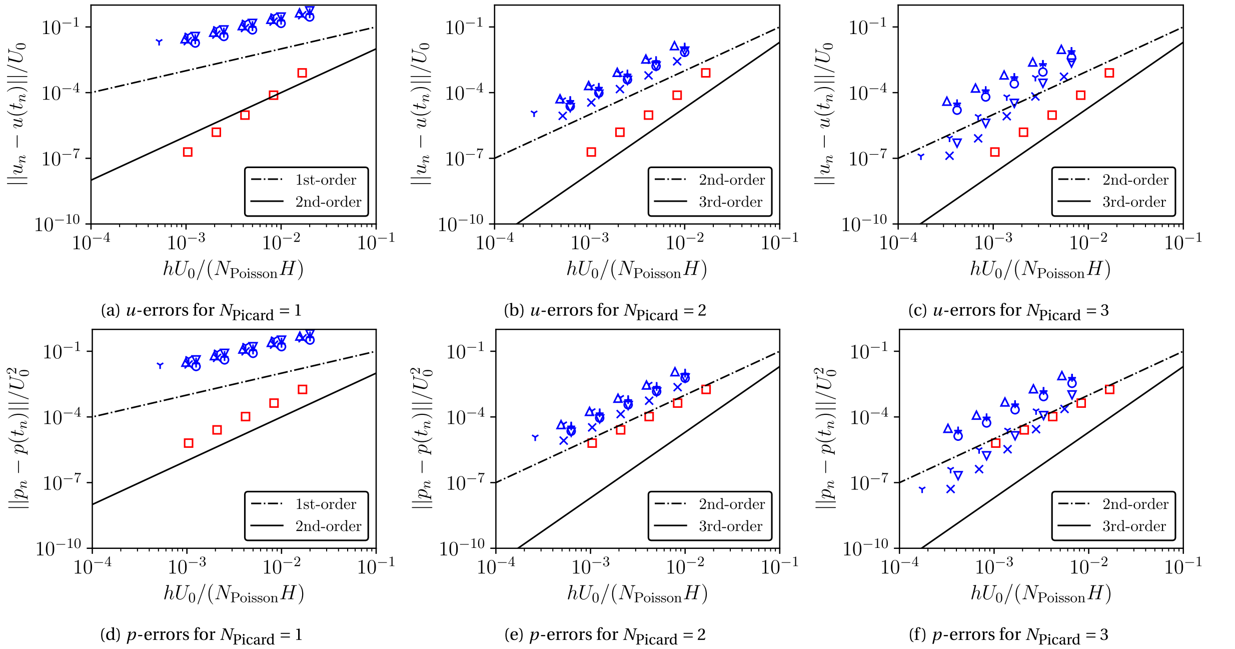
Method	s_{eff}	250 × 40 grid		2500 × 400 grid	
		$U_0 h_{\max}/H$	$U_0 h_{\max}/s_{\text{eff}} H$	$U_0 h_{\max}/H$	$U_0 h_{\max}/s_{\text{eff}} H$
2-step BDF	1	1/32	1/32	1/100	1/100
3-step BDF	1	1/25	1/25	1/100	1/100
1-stage Gauss	1	1/25	1/25	1/100	1/100
2-stage SDIRK	2	2/25	1/25	1/50	1/100
3-stage SDIRK	3	1/20	1/60	1/100	1/300
3-stage ESDIRK	2	2/25	1/25	1/50	1/100
4-stage ESDIRK	3	1/10	1/30	2/125	2/375
3-stage HERK	3	1/20	1/60	1/250	1/750

A preliminary stability analysis of the eight methods under consideration is conducted by evaluating their maximum allowable time-step sizes ($h_{\max}$). Each method is applied to simulate the BFS flow with an increasing sequence of time-step sizes. We verify whether each simulation runs to completion or diverges at a given time-step size to determine $h_{\max}$ (see Table 5.15). To compare the stability of the methods, $h_{\max}$ for each is divided by the effective number of intermediate stages, s_{eff}. Based on this criterion, the 3-step BDF method, 1-stage Gauss method, 2-stage SDIRK method, and 3-stage ESDIRK methods demonstrates similar stability, superior to the remaining methods, which aligns with their relatively large c_{ssp}. On the coarser grid, the ratio $h_{\max}/s_{\text{eff}}$ for the optimal implicit methods is only 2.4 times that of the 3-stage HERK method. However, on the finer grid, where the semi-discrete Navier-Stokes system becomes much stiffer, this ratio increases to 7.5. It is notable that the 1-stage Gauss method does not maintain its optimal $h_{\max}/s_{\text{eff}}$ on the finer grid when $N_{\text{Picard}} = 1$, indicating weaker linear stability compared to the L-stable SDIRK and ESDIRK methods. It is also noteworthy that for the implicit methods, reducing the grid spacing by a factor of ten does not lead to a proportional reduction in $h_{\max}$ on the finer grid. In contrast, the $h_{\max}$ for the 3-stage HERK method on the finer grid decreases to below one-tenth of its value on the coarser grid. This suggests that as the stiffness of the semi-discrete Navier-Stokes system increases, the linear stability of the time-discretization method becomes more important than its nonlinear stability.

Next, the divergence-free constraint is examined for the 1-stage Gauss method. The other implicit Runge-Kutta (IRK) methods in this experiment are stiffly accurate, ensuring $u_{n+1} = U_{ns}$. Thus, $g(t_{ns}, U_{ns}) = g(t_{n+1}, u_{n+1}) = 0$ is directly enforced in solving the final-stage intermediate equations, see eqs. (3.8), (3.4c) and (3.4d). In contrast, the 1-stage Gauss method indirectly enforced $\overline{g(t_{n+1}, u_{n+1})} = 0$ through the perturbed intermediate boundary velocity within the IRK-CP scheme. Fig. 5 shows the time histories of $h\overline{|g(t_n, v_n)|}$ (where the overline represents a cell-volume-based weighted average) for the 1-stage Gauss method at different time-step sizes on the finer grid, all of which approach zero. This validates the IRK-CP scheme's ability to enforce the divergence-free constraint for non-stiffly accurate methods.

Finally, temporal errors for each method are evaluated across various time-step sizes to investigate convergence. Since the exact solution of the semi-discrete Navier-Stokes system is required to quantify these errors, it is approximated using the 4-stage ESDIRK method with a dimensionless time-step size of $hU_0/H = 1 \times 10^{-4}$. We use time-step sizes from the sets $\{h_{\max}/2^4, \cdots, h_{\max}/2^0\}$ for the coarser grid and $\{h_{\max}/2^2, h_{\max}/2^1, h_{\max}/2^0\}$ for the finer grid. Smaller time-step sizes are not considered, as the temporal errors would approach the tolerance level for linear solvers. Implicit methods also employ N_{Picard} ranging from 1 to 3 for nonlinearity correction. To assess efficiency, the maximum temporal errors, determined at five uniformly distributed time instants over the interval $t \in [0, T]$, are plotted against the dimensionless time-step size $hU_0/(N_{\text{Poisson}} H)$ (see Figs. 6 and 7). This allows comparison of errors at specific $hU_0/(N_{\text{Poisson}} H)$ values, where lower errors indicate higher efficiency. For both grids, the IRK-CP scheme enables both differential and algebraic components (velocity and pressure) to achieve the optimal convergence order based on the employed implicit Runge-Kutta coefficients, provided sufficiently small residual errors are attained. When $N_{\text{Picard}} = 1$, all seven implicit methods achieve only 1st-order convergence, with the 1-stage Gauss method performing slightly better in efficiency. When $N_{\text{Picard}} = 2$, all implicit methods reach at least 2nd-order convergence, with the 4-stage ESDIRK method proving to be the most efficient. Increasing N_{Picard} to 3 allows the 3-step BDF method, 3-stage SDIRK method, and 4-stage ESDIRK method to recover 3rd-order convergence, making them significantly more efficient than the other four 2nd-order implicit methods across the entire range of time-step sizes.

The 3-stage HERK method also attains its expected convergence orders. On both grids, it demonstrates significantly higher efficiency for velocity than any of the implicit methods. Although its efficiency advantage diminishes slightly on the finer grid, it still requires roughly half the computational effort of the 3rd-order implicit methods to achieve comparable error results. As for pressure,



(a) u-errors for $N_{\text{Picard}} = 1$

(b) u-errors for $N_{\text{Picard}} = 2$

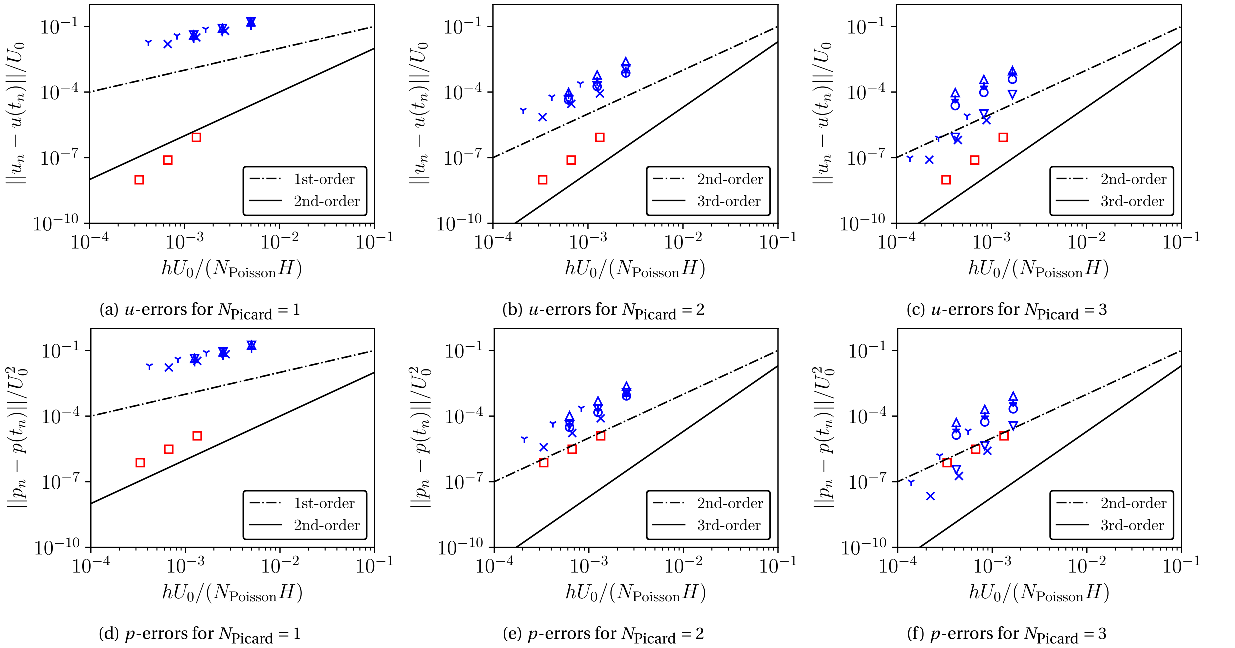
(c) u-errors for $N_{\text{Picard}} = 3$

(d) p-errors for $N_{\text{Picard}} = 1$

(e) p-errors for $N_{\text{Picard}} = 2$

(f) p-errors for $N_{\text{Picard}} = 3$

Fig. 6. Dimensionless global errors of selected methods for the numerical simulation of the BFS flow with unsteady inflow on the 250 × 40 grid: 2-step BDF △; 3-step BDF ▽; 2-stage SDIRK ⅄; 3-stage SDIRK Y; 3-stage ESDIRK +; 4-stage ESDIRK ×; 3-stage HERK □.



(a) u-errors for $N_{\text{Picard}} = 1$

(b) u-errors for $N_{\text{Picard}} = 2$

(c) u-errors for $N_{\text{Picard}} = 3$

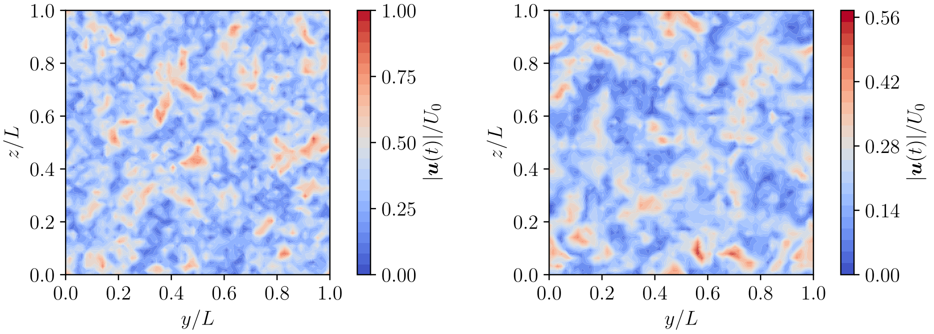
(d) p-errors for $N_{\text{Picard}} = 1$

(e) p-errors for $N_{\text{Picard}} = 2$

(f) p-errors for $N_{\text{Picard}} = 3$

Fig. 7. Dimensionless global errors of selected methods for the numerical simulation of the BFS flow with unsteady inflow on the 2500 × 400 grid: 2-step BDF △; 3-step BDF ▽; 2-stage SDIRK ⅄; 3-stage SDIRK Y; 3-stage ESDIRK +; 4-stage ESDIRK ×; 3-stage HERK □.

all 3rd-order IRK methods outperform this HERK method due to one order higher convergence for pressure. Under the same time-step size, the 4-stage ESDIRK method, using $N_{\text{Picard}} = 3$ and $N_{\text{PISO}} = 2$, is approximately 6 times more computationally expensive than the 3-stage HERK method. Despite the 4-stage ESDIRK method's advantage in terms of maximum allowable time-step size on the finer grid, it is insufficient to match the efficiency of the 3-stage HERK method for velocity. To make implicit methods competitive to explicit methods of the same order in terms of efficiency, techniques for accelerating the iterative solution of the nonlinear equations governing velocity and pressure must be developed.



(a) the synthesis generator

(b) the precursor simulation

Fig. 8. Magnitude contour of the turbulent velocity field from the synthesis generator and the precursor simulation.

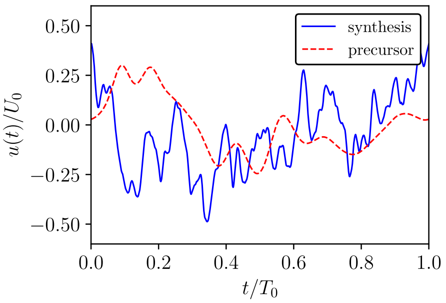


Fig. 9. Time histories of a one-point streamwise turbulence from the synthesis generator and the precursor simulation.

5.3. Decaying of homogeneous isotropic turbulence

In the final numerical experiment, the temporal convergence of the IRK-CP scheme is further validated through the large eddy simulation (LES) of decaying homogeneous isotropic turbulence (HIT). Diagonally implicit RK coefficients are employed, and the HIT is injected into a box computational domain with periodic boundary conditions in the cross-streamwise directions. The turbulent velocity field at the domain's inlet is obtained from a precursor simulation. The precursor simulation itself imposes the HIT generated from an energy-spectrum-based synthetic method as proposed in [54] at the inlet. The precursor simulation is carried out in a domain of dimensions 0.64m×0.32m×0.32m with a grid resolution of 128×64×64 in the longitudinal, vertical, and lateral directions, respectively. For the main simulations used to evaluate the temporal convergence of the time discretization methods, the longitudinal dimension and grid size are halved. Both precursor and main simulations use a zero-gradient velocity condition at the outlet, while the pressure field employs a zero-gradient condition at the inlet and a fixed-zero value at the outlet. The subgrid-scale turbulence is modeled using the classical Smagorinsky model.

The energy spectrum for the HIT, required by the synthetic method, is based on grid-generated turbulence as experimentally studied in [55]. Specifically, the energy spectrum of turbulence sampled at a dimensionless distance $U_0t/M = 42$ downstream of a grid (where $U_0 = 10$ m/s is the upstream airspeed and $M = 0.0508$ m denotes the grid's mesh size in this section) is considered. The HIT is synthesized in a cubic domain with dimensions $L = 0.32$ m and a grid resolution of $8192 \times 32 \times 32$. The turbulence is then converted into a discrete-time velocity field (see Fig. 8a) on a 32×32 plane grid over 8192 time steps, assuming a mean advection velocity of $U_0 = 1$ m/s in the precursor simulation. The time step for this velocity field is $h_0 = 1/25600$ seconds, with a time period of $T_0 = 0.32$ seconds. The time-periodicity of the synthetic HIT allows for extending the time length by duplicating periods as needed.

The precursor simulation runs for 1.88 seconds, using the same time step as the synthetic velocity field. Starting from $t = 1.28$ seconds, the velocity field is recorded on a slice plane located 0.285 m downstream of the inlet. This provides a new discrete-time velocity field (see Fig. 8b) a time length of $T = 0.6$ seconds and a uniform time step size of h_0. A comparison between the time-histories of the streamwise velocity component from the synthetic method and the precursor simulation is presented in Fig. 9. Additionally, the energy spectrum function of the precursor HIT is compared to that of the HIT sampled downstream of the grid at $U_0t/M = 98$ in [55], see Fig. 10. The time it takes for the grid-generated HIT to travel from $U_0t/M = 42$ to 98 is approximately 0.285 seconds, which matches the time for the synthetic HIT to reach the sampling slice in the precursor simulation.

Next, the main simulations, with a total time of $T = 0.6$ seconds, are performed using the turbulent velocity field from the precursor simulation as inflow. The simulations examine the convergence of all considered time discretization methods. For initial conditions, a uniform vector value $(1, 0, 0)$ m/s and a zero value are assigned to u_0 and p_0, respectively. As $v'(t_0) \neq 0$, the consistency condition (2.2) is apparently not satisfied. Ideally, p_0 should be computed using eq. (1.3), but this is impractical, as $v'(t_0)$ is unavailable in most engineering applications. As a result, convergence issues arise for the Gauss, ESDIRK and HERK methods, as their approximations for

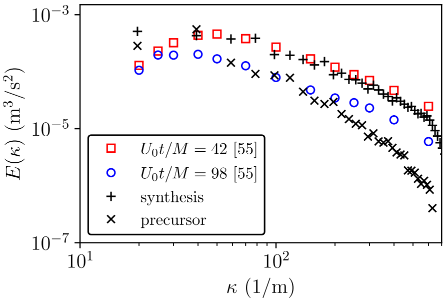
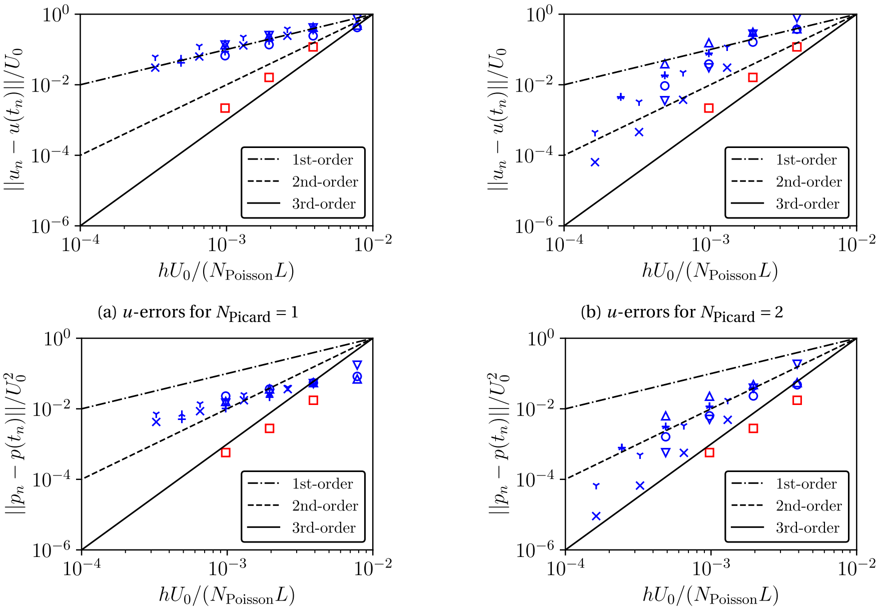


Fig. 10. A comparison of the energy spectrum function for the considered HIT from different approaches.



(a) u-errors for $N_{\text{Picard}} = 1$

(b) u-errors for $N_{\text{Picard}} = 2$

(c) p-errors for $N_{\text{Picard}} = 1$

(d) p-errors for $N_{\text{Picard}} = 2$

Fig. 11. Dimensionless global errors of selected methods for the LES of the decaying HIT from a precursor simulation: 2-step BDF △; 3-step BDF ▽; 1-stage Gauss ○; 2-stage SDIRK ⅄; 3-stage SDIRK Y; 3-stage ESDIRK +; 4-stage ESDIRK ×; 3-stage HERK □.

(u_{n+1}, p_{n+1}) depend on both u_n and p_n. In contrast, stiffly accurate SDIRK methods, whose approximations depend only on u_n, avoid this issue. A practical remedy for those methods is to use a stiffly accurate SDIRK method (the 3-stage SDIRK method is applied here) for the first few time steps before switching to these methods.

Due to the fixed time intervals of the imposed turbulent inflow, the selection of time step sizes is less flexible than in the previous experiment. For this case, the IRK methods based on the IRK-CP scheme are tested with time step sizes as powers of 2 multiplied by h_0. Temporal errors for all implicit methods are evaluated for the same set of time step sizes $\{2^4 h_0, 2^5 h_0, 2^6 h_0, 2^7 h_0\}$. The HERK, which possesses coefficients $c_i \in \{0, 1/3, 1\}$, requires time steps that are integer multiples of $3h_0$. Its temporal errors are therefore evaluated for time step sizes in the set $\{3 \times 2^3 h_0, 3 \times 2^4 h_0, 3 \times 2^5 h_0\}$. The exact solution, used for error estimation, is approximated using the 3-step BDF method with a time step size of h_0. We employ $N_{\text{PISO}} = 2$, and N_{Picard} ranges from 1 to 2 when implementing Algorithm 1. The maximum temporal errors, sampled from five uniformly distributed time instants over $t \in [0, T]$, are plotted against a dimensionless time step size $hU_0/(N_{\text{Poisson}}L)$ in Fig. 11. Similar to the previous experiment, all implicit methods exhibit only first-order convergence when $N_{\text{Picard}} = 1$. Increasing N_{Picard} to 2 restores the expected convergence rates, with the 4-stage ESDIRK method emerging as the most efficient implicit method. Again, the 3-stage HERK method is more efficient than the 4-stage ESDIRK method for velocity. But its efficiency advantage is not as significant as the previous experiment since the 4-stage ESDIRK method already attains 3rd-order convergence on the second Picard iteration loop.

6. Concluding remarks

This study examines the convergence of IRK methods for the incompressible N-S equations with unsteady inflow. The convergence analysis reveals that the IRK-DAE1 scheme, which treats the problem as an index-1 DAE system, is more efficient in achieving higher orders of convergence. The SRK-DAE2 scheme, derived from the application of SRK methods for general index-2 DAE systems, ensures the exact satisfaction of eq. (1.5b) while matching the convergence of the IRK-DAE1 scheme for Gauss and Radau IA methods. In contrast, the IRK-DAE2 scheme, which solves the problem as an index-2 system, experiences varying degrees of order reduction.

Considering the limitations of the three examined schemes in terms of numerical efficiency, satisfaction of algebraic constraints, or orders of convergence, we enhance the performance of IRK methods by modifying the original formula of the IRK-DAE1 scheme with constrained perturbations. These perturbations essentially alter the values of unsteady inflow velocities in the fully-discrete N-S equations of RK intermediate stages (see eqs. (3.23) and (3.24)). The resulting IRK-CP scheme (3.3) achieves the same orders of convergence as the IRK-DAE1 scheme while ensuring the exact satisfaction of the continuity equation for both stiffly accurate and non-stiffly accurate RK coefficients. In applying the IRK-CP scheme, having an explicit expression for the time-varying inflow velocity $v(t)$ is an ideal, yet restrictive, requirement. If this is not feasible, it is sufficient to have access to the velocity values at each time step and some intermediate time instants between successive steps.

The convergence of the IRK-CP scheme is initially derived mathematically and subsequently verified through three numerical experiments. Based on the IRK-CP scheme presented in this paper and its convergence results, we demonstrate the feasibility of obtaining pressure with the same order of accuracy as velocity using IRK methods, without the additional solution of eq. (1.3), even if the boundary condition for velocity fields is time-varying. It is crucial to emphasize that this statement is valid only when hypotheses H1 and H2 hold for the semi-discrete N-S system eq. (1.5). Among various IRK methods considered in this study, we find that stiffly accurate DIRK methods benefit most from the IRK-CP scheme, since their conventional implementation via the IRK-DAE2 scheme can only achieve second-order accuracy for pressure at best.

This study investigates the performance of various methods in two practical applications: a two-dimensional backward-facing step flow with unsteady inflow and decaying homogeneous isotropic turbulence. The methods include the 1-stage Gauss method (i.e., the implicit midpoint rule), 2-stage and 3-stage stiffly accurate SDIRK and ESDIRK methods, alongside two BDF methods and a 3-stage HERK method, for comparison on stability and efficiency. Results show that higher-order methods require smaller time-step sizes and solution residuals (for implicit methods) to outperform lower-order methods in efficiency. Among the 2nd-order implicit methods, the 1-stage Gauss method, benefiting from the IRK-CP scheme's convergence order for pressure and enforcement of the divergence-free constraint, exhibits optimal efficiency. Additionally, the 2nd-order SDIRK and ESDIRK methods offer improved stability and efficiency over the 2-step BDF method, while maintaining a low-storage requirement of 2 memory registers for velocity and 1 for pressure. In contrast, the 2-step BDF method requires 3 registers for velocity and 1 for pressure, and the 1-stage Gauss method needs 2 for both. The 3rd-order ESDIRK method, using the same memory as the 2-step BDF, surpasses the 3-step BDF in efficiency. Given that these IRK methods only need minor modifications to existing code, integrating them into open-source CFD software like OpenFOAM is highly recommended.

In both numerical experiments, the considered 2nd- or 3rd-order IRK and BDF methods are not as efficient as the HERK method (of 3rd-order for velocity and 2nd-order for pressure) in terms of the solution efficiency for velocity. Even in a semi-discrete Navier-Stokes system, which exhibits greater stiffness due to finer grid spacing, the implicit methods' advantage in allowable time-step sizes is not sufficient to offset the computational cost of solving the nonlinear equations governing velocity and pressure. For the solution efficiency for pressure, the 3rd-order IRK and BDF methods outperform the HERK method due to a higher convergence order for pressure. The efficiency comparison in this study indicates that higher-order HERK methods are most efficient for the accurate time integration of a given semi-discrete Navier-Stokes system. It is crucial to note that this conclusion should be generalized with caution, as factors such as mesh grid, code, and turbulence models can influence the results. Nevertheless, HERK methods still suffer from severe order reduction due to space-time errors and relatively high memory consumption. Moreover, if iterative acceleration techniques for implicit methods can be developed, they may become competitive to HERK methods of the same order in terms of temporal solution efficiency.

Another notable finding of this study is the order of local and global errors of algebraic components in the SRK-DAE2 scheme of eq. (2.9) for general semi-explicit index-2 problems (D.1). Since the original study by [23], only the case where algebraic components are obtained from the solution of an implicit equation regarding differential components has been considered. Suggesting that a less computationally demanding approach using eq. (D.3b) should be investigated independently. From Theorems 2 and 3, we find that SRK methods achieve higher orders for algebraic components in general index-2 problems compared to conventional IRK methods when using Gauss and Radau IA types of RK coefficients.

CRediT authorship contribution statement

Yunzhu Cai: Writing – review & editing. **Jiawei Wan:** Writing – original draft, Methodology, Conceptualization. **Ahsan Kareem:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is supported in part by funds from the Natural Science Foundation of Jiangsu Province of China (Grant No. BK20230337), the Natural Science Research of Jiangsu Higher Education Institutions of China (Grant No. 22KJB560004), and the Robert M. Moran Professorship at University of Notre Dame.

Appendix A. Convergence of the IRK-DAE1 scheme (2.3)

The convergence of the IRK-DAE1 scheme (2.3) should be determined by considering eq. (2.1) as a general index-1 problem since $f(t,u,p)$ is non-linear in t and u. Consequently, the order of the global error for the u-component given by eq. (2.3) is identical to the order of a conventional RK method with the same coefficients (a_{ij}, b_i, and c_i) for ODEs, which will be denoted by κ hereafter, i.e.,

$$u_n - u(t_n) = O(h^{\kappa}). \tag{A.1}$$

From Theorem 5.1 in Chapter IV of [13], we have $\kappa \geq \min(\eta, \zeta+\xi+1, 2\zeta+2)$ for methods satisfying $B(\eta)$, $C(\zeta)$, and $D(\xi)$. According to Theorem 3.1 of [12], and assuming that $\kappa \geq \zeta+1$, the global error for the p-component computed from eq. (2.7), for IRK methods of type I, admits the estimate

$$p_n - p(t_n) = \begin{cases} O(h^{\zeta+1}) & \text{if } -1 \leq R(\infty) < 1, \\ O(h^{\zeta}) & \text{if } R(\infty) = 1, \\ O(h^{\kappa}) & \text{if } b_i = a_{si} \text{ for all } i. \end{cases} \tag{A.2}$$

When the p-component is obtained from the solution of eq. (2.6) instead, its global error has the same order of convergence as that of the u-component, for IRK methods of type I and type II both.

Appendix B. Convergence of the IRK-DAE2 scheme (2.8)

According to Lemma 4.3, Theorem 4.4, and 4.9 of [12] concerning general index-2 DAEs, and considering the fact that hypothesis H3 does not improve the convergence results predicted by those theorems, the global error for the u-component attained by the IRK-DAE2 scheme (2.8), for general IRK methods of type I, is of the magnitude

$$u_n - u(t_n) = \begin{cases} O(h^{\min(\eta,\zeta+1)}) & \text{if } -1 \leq R(\infty) < 1 \text{ and } \zeta \geq 1, \\ O(h^{\min(\eta,\zeta)}) & \text{if } R(\infty) = 1 \text{ and } \zeta \geq 2. \end{cases} \tag{B.1}$$

For stiffly accurate IRK methods of type I, the application of Theorem 5.10 in Chapter VII of [13] and hypothesis H3 indicates that the global error for the u-component can be sharpened with

$$u_n - u(t_n) = O(h^{\min(\eta,2\zeta+1,\zeta+\xi+1)}). \tag{B.2}$$

For IRK methods of type II, Theorem 5.2 in [33] suggests

$$u_n - u(t_n) = \begin{cases} O(h^{\min(\eta,2\zeta+1,\zeta+\xi+1)}) & \text{if } -1 \leq R(\infty) < 1, \\ O(h^{\min(\eta,2\zeta,\zeta+\xi+1)}) & \text{if } R(\infty) = 1. \end{cases} \tag{B.3}$$

The convergence order of the p-component depends on which of the above-mentioned formulas is employed for the determination of p_{n+1}. When eq. (2.6) is selected, the global error for the p-component has the same order as that of the u-component given by eq. (B.1) or eq. (B.2). When eq. (2.7) is chosen, Theorem 4.6 and 4.9 of [12] suggest that IRK methods of type I admit the following error estimate

$$p_n - p(t_n) = \begin{cases} O(h^{\min(\eta,\zeta)}) & \text{if } |R(\infty)| < 1, \\ O(h^{\min(\eta,\zeta-1)}) & \text{if } R(\infty) = -1 \text{ and } \zeta \geq 1, \\ O(h^{\min(\eta,\zeta-2)}) & \text{if } R(\infty) = 1 \text{ and } \zeta \geq 2. \end{cases} \tag{B.4}$$

For IRK methods of type II using $p_{n+1} = P_{ns}$, Theorem 5.2 in [33] gives

$$p_n - p(t_n) = \begin{cases} O(h^{\min(\eta,\zeta)}) & \text{if } -1 \leq R(\infty) < 1, \\ O(h^{\min(\eta,\zeta-1)}) & \text{if } R(\infty) = 1. \end{cases} \tag{B.5}$$

It is noteworthy that the error estimates of eqs. (B.4) and (B.5), for the index-2 system (1.5) satisfying hypotheses H1 to H3, are identical to those of a general index-2 system, since hypothesis H3 does not reduce enough order conditions for the p-component so as to make a difference. Moreover, the error estimates of eqs. (B.4) and (B.5) do not necessarily give the optimal convergence of a particular method for the p-component. For example, stiffly accurate DIRK methods, which only satisfy $C(1)$, can actually attain 3rd-order for differential components and 2nd-order for the algebraic components of general index-2 DAEs, when their coefficients fulfill $b^T A^{-2} c^2 = 2$ in addition to the classical order conditions of order 3 for ODEs. Another example would be stiffly accurate ESDIRK methods, which satisfy $C(2)$ at most, can actually attain 3rd-order for the algebraic components of general index-2 DAEs by fulfilling $\tilde{b}^T \tilde{A}^{-2} \tilde{c}^3 = 3$ in addition to the classical order conditions of order 3 for ODEs.

Appendix C. Convergence of the SRK-DAE2 scheme (2.9)

From Theorem 4.4 and 4.5 of [23] and using hypothesis H3, the global error for the u-component resulting from the SRK-DAE2 scheme (2.9) is of the magnitude

$$u_n - u(t_n) = \begin{cases} O(h^{\min(\eta,2\zeta+1,\zeta+\min(\xi,s-1)+1)}) & \text{if } \xi \leq s-2, \\ O(h^{\min(\eta,2\max(\zeta,\eta-s+1)+1,2\zeta+3,\zeta+\min(\xi,s-1)+1)}) & \text{if } \xi \geq s-1. \end{cases} \tag{C.1}$$

When p_{n+1} is determined from eq. (2.6), the global error for the p-component shares the same order as that of the u-component. The case in which p_n is computed with eq. (2.7) requires further investigation, as this issue is not addressed in the original work of [23]. In Appendix D, we present two theorems concerning the local and global errors for algebraic components such as p in this case. According to Theorems 2 and 3, the p-component computed with eq. (2.7) satisfies

$$p_n - p(t_n) = \begin{cases} O(h^{\min(\eta,\zeta)}), & \text{if } -1 \leq R(\infty) < 1 \text{ and } \xi < s-1, \\ O(h^{\min(\eta+1,\zeta+1,\max(\min(\eta,\zeta),\eta-s+1))}), & \text{if } -1 \leq R(\infty) < 1 \text{ and } \xi \geq s-1, \\ O(h^{\min(\eta,\zeta)-1}), & \text{if } R(\infty) = 1 \text{ and } \xi < s-1, \\ O(h^{\min(\eta,\zeta,\max(\min(\eta,\zeta),\eta-s+1)-1)}), & \text{if } R(\infty) = 1 \text{ and } \xi \geq s-1. \end{cases} \tag{C.2}$$

Based on the above error estimate, one can easily verify that the SRK-DAE2 scheme attains higher orders of convergence for the p-component than the IRK-DAE2 scheme in terms of Gauss and Radau IA methods.

Appendix D. Theories on the order conditions of algebraic components in SRK methods

Theorem 2. *Consider a general semi-explicit index-2 system of DAEs given by*

$$y' = f(y,z), \quad g(y) = 0, \tag{D.1}$$

and suppose that the initial conditions are consistent, i.e.,

$$g(y_0) = 0, \quad (g_y f_z)(y_0, z_0) = 0, \tag{D.2}$$

and that the inverse of $(g_y f_z)(y,z)$ exists and is bounded in a neighborhood of the exact solution of the system. Given an s-stage SRK method with the following form:

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i f(Y_{ni}, Z_{ni}), \tag{D.3a}$$

$$z_{n+1} = z_n + \sum_{i,j=1}^{s} b_i \omega_{ij}(Z_{nj} - z_n), \tag{D.3b}$$

$$Y_{ni} = y_n + h\sum_{j=1}^{s} a_{ij} f(Y_{nj}, Z_{nj}), \quad 0 = \sum_{j=1}^{s} \tilde{\omega}_{ij} g(Y_{nj}) + \tilde{\omega}_{i,s+1} g(y_{n+1}), \tag{D.3c}$$

with a non-singular coefficient matrix $[a_{ij}]$, and satisfying the conditions of $B(\eta)$ and $C(\zeta)$. Then, the local error of the z-component given by the method satisfies

$$\delta^z(y,z,h) = O(h^{\min(\eta,\zeta)}). \tag{D.4}$$

If the condition of $D(s-1)$ is additionally satisfied, the above error estimate can be sharpened with

$$\delta^z(y,z,h) = O(h^{\min(\eta+1,\zeta+1,\max(\min(\eta,\zeta),\eta-s+1))}). \tag{D.5}$$

Proof. This theorem can be proven based on the tree and elementary differential theories [12,13] developed for semi-explicit differential-algebraic systems of index 2. First, according to Theorem 4.2 of [23], the Z_{ni} in eq. (D.3) satisfy

$$Z_{ni}^{(\zeta)}|_{h=0} = \sum_{\substack{u \in LDAT2_z \\ \rho(u)=\zeta}} \gamma(u)\Phi_i(u)F(u)(y_n, z_n), \tag{D.6}$$

with the coefficients $\Phi_i(u)$ given recursively by $\Phi_i(\tau) = 1$ and

$$\Phi_i(t) = \sum_{\mu_1=1}^{s} \cdots \sum_{\mu_m=1}^{s} \alpha_{i\mu_1} \cdots \alpha_{i\mu_m} \Phi_{\mu_1}(t_1) \cdots \Phi_{\mu_m}(t_m)\Phi_i(u_1) \cdots \Phi_i(n) \text{ for } t = \left[t_1, \ldots, t_m, u_1, \ldots, u_n\right]_y, \tag{D.7a}$$

$$\Phi_i(u) = \sum_{j=1}^{s+1} \sum_{\mu_1=1}^{s} \cdots \sum_{\mu_m=1}^{s} \tilde{\omega}_{ij}\alpha_{j\mu_1} \cdots \alpha_{j\mu_m} \Phi_{\mu_1}(t_1) \cdots \Phi_{\mu_m}(t_m) \text{ for } u = \left[t_1, \ldots, t_m\right]_z, \tag{D.7b}$$

where the notation τ, $t \in LDAT2_y$, $u \in LDAT2_z$, $F(u)$, $\rho(u)$, and $\gamma(u)$ are identical to those of Theorem 5.6 in Chapter VII of [13]. Following from eqs. (D.3b) and (D.6), we obtain

$$z_{n+1}^{(\zeta)}|_{h=0} = \sum_{\substack{u\in LDAT2_z \\ \rho(u)=\zeta}} \gamma(u) \sum_{i,j=1}^{s} b_i\omega_{ij}\Phi_j(u)F(u)(y_n, z_n). \tag{D.8}$$

Based on eq. (D.8), the local error for the z-component given by the method eq. (D.3) satisfies

$$z_{n+1} - z(t_n + h) = O(h^{\mu}), \tag{D.9}$$

if and only if

$$\sum_{i,j=1}^{s} b_i\omega_{ij}\Phi_j(u) = \frac{1}{\gamma(u)} \text{ for } u \in DAT2_z \text{ with } \rho(u) \le \mu - 1. \tag{D.10}$$

When $B(\eta)$ and $C(\zeta)$ are satisfied, we have

$$b^T c^{k-1} = \frac{1}{k} \text{ for } k = 1, \dots, \eta, \tag{D.11a}$$

$$A^{-1}c^k = kc^{k-1} \text{ for } k = 1, \dots, \zeta, \tag{D.11b}$$

and consequently, the order conditions given by eq. (D.10) reduce to

$$b^T A^{-1}\Omega \begin{pmatrix} c^k \\ 1 \end{pmatrix} \text{ for } k = 1, \dots, \mu = \min(\eta, \zeta). \tag{D.12}$$

Multiplying b^T on both sides of eq. (D.11b) and using eq. (D.11a), we also have, for $k = 1, \dots, \min(\eta, \zeta) + 1$, the equality

$$b^T A^{-1}c^{k-1} = 1. \tag{D.13}$$

Meanwhile, Theorem 3.2 of [23] suggests that, for $k = 1, \dots, \min(\eta, \zeta)$, the equality

$$\Omega \begin{pmatrix} c^k \\ 1 \end{pmatrix} = kc^{k-1}, \tag{D.14}$$

holds on the conditions of $B(\eta)$ and $C(\zeta)$. If the condition of $D(s-1)$ is also satisfied, eq. (D.14) holds for $k = 1, \cdots, \max(\min(\eta, \zeta), \eta - s + 1)$. Equations (D.13) and (D.14) together ensure the satisfaction of eq. (D.12), which proves the statement of the theorem. □

Theorem 3. *Consider the index-2 problem (D.1) with consistent initial values, and suppose that the inverse of $(g_y f_z)(y, z)$ exists and is bounded in a neighborhood of the exact solution of the system. Given an s-stage SRK method (D.3) with a non-singular coefficient matrix $[a_{ij}]$ and satisfying the conditions of $B(\eta)$, $C(\zeta)$, and $|R(\infty)| \le 1$. If the global error for the y-component and the local error for the z-component are both $O(h^k)$ in the method, then the global error for the z-component satisfies*

$$z_n - z(t_n) = \begin{cases} O(h^k), & \text{if } -1 \le R(\infty) < 1, \\ O(h^{k-1}), & \text{if } R(\infty) = 1. \end{cases} \tag{D.15}$$

Proof. A proof of the theorem, for $-1 < R(\infty) \le 1$, can be obtained following the lines of Theorem 4.6 and 4.9 of [12], the details of which are therefore omitted. For the case in which $R(\infty) = -1$, an asymptotic expansion regarding $z_n - z(t_n)$ should be considered instead. Following the proof of Theorem 4.8 of [12], it can be shown that the Z_{ni} in the SRK method eq. (2.9) possesses an asymptotic expansion of the form

$$Z_{ni} = z(t_{ni}) + h^k B_{i,k}(t_n) + \cdots + h^N B_{i,N}(t_n) + O(h^{N+1}), \tag{D.16}$$

where $B_{i,k}$ $(i = k, \cdots, N)$ are expansion coefficients. Meanwhile, the difference between z_n and $z(t_n)$ admits the recurrence relation

$$z_{n+1} - z(t_{n+1}) = R(\infty)\left(z_n - z(t_n)\right) + \sum_{i,j=1}^{s} b_i\omega_{ij}\left(Z_{nj} - z(t_{nj})\right) + O(h^{\eta+1}) + O(h^{\zeta+1}), \tag{D.17}$$

for RK coefficients satisfying $B(\eta)$ and $C(\zeta)$. Inserting eq. (D.16) into eq. (D.17) and considering $k \le \min(\eta, \zeta) + 1$ according to Theorem 3, we obtain

$$z_{n+1} - z(t_{n+1}) = R(\infty)\left(z_n - z(t_n)\right) + \delta_n, \tag{D.18}$$

with δ_n admitting an asymptotic expansion of the form

$$\delta_n = h^k d_k(t_n) + \cdots + h^N d_N(t_n) + O(h^{N+1}). \tag{D.19}$$

The order of $z_n - z(t_n)$ for $R(\infty) = -1$ then follows in exactly the same way as in the proof of Theorem 3.2 of [12]. □

Appendix E. Proof of Theorem 1

Proof. The essence of Theorem 1 is that the perturbations introduced in the proposed IRK-CP scheme do not reduce the order of convergence compared to its unperturbed counterpart, the IRK-DAE1 scheme. The proof proceeds in three main steps: First, we estimate the differences between the intermediate values of the perturbed and unperturbed schemes within a local step. Second, we use this estimate to derive the differences between the local solutions of the perturbed and unperturbed schemes. Finally, we analyze how these differences propagate and accumulate throughout the solution domain.

The proof of Theorem 1 incorporates several established results. Specifically, Theorems 3.1 and 3.2 from [12], which prove the convergence of the IRK-DAE1 scheme without perturbations, are used directly. Lemma 5 in [56], concerning the existence of a locally unique solution for a general perturbed scheme, and Lemma 6 in [56], which estimates the differences between intermediate values of perturbed and unperturbed schemes within a local step, are also employed. Furthermore, the technique for investigating the propagation and accumulation of local differences follows that of Theorem 7 in [56]. Lastly, Lemma 2, which provides the asymptotic expansion for the global error of the z-component in the IRK-CP scheme, is a slight adaptation of Theorem 3.2 from [12]. To apply these theorems and lemmas, it is advantageous to treat eq. (2.1) as a general semi-explicit index-1 system

$$y' = f(y,z), \quad g(y,z) = 0, \tag{E.1}$$

where f and g are sufficiently differentiable, and the inverse of $g_z(y,z)$ (the subscript in g_z represents the partial derivative of g with respect to z) exists and is bounded. The primary focus is on IRK methods of type I, in which the matrix $[a_{ij}]$ is presumed to be non-singular.

For the solution of eq. (E.1), the IRK-DAE1 scheme, in conjunction with eq. (2.7) employed for approximating the algebraic components, can be expressed as follows

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i f(Y_{ni}, Z_{ni}), \tag{E.2a}$$

$$z_{n+1} = z_n + \sum_{i,j=1}^{s} b_i \omega_{ij}\left(Z_{nj} - z_n\right), \tag{E.2b}$$

$$Y_{ni} = y_n + h\sum_{j=1}^{s} a_{ij} f(Y_{nj}, Z_{nj}), \quad g(Y_{ni}, Z_{ni}) = 0. \tag{E.2c}$$

Upon application to index-1 DAEs of eq. (E.1), the IRK-CP scheme (3.3) is transformed into

$$\hat{y}_{n+1} = \hat{y}_n + h\sum_{i=1}^{s} b_i f(\hat{Y}_{ni}, \hat{Z}_{ni}) + h\delta_{n+1}, \tag{E.3a}$$

$$\hat{z}_{n+1} = \hat{z}_n + \sum_{i,j=1}^{s} b_i \omega_{ij}\left(\hat{Z}_{nj} - \hat{z}_n\right), \tag{E.3b}$$

$$\hat{Y}_{ni} = \hat{y}_n + h\sum_{j=1}^{s} a_{ij} f(\hat{Y}_{nj}, \hat{Z}_{nj}) + \delta_{ni}, \quad g(\hat{Y}_{ni}, \hat{Z}_{ni}) + \theta_{ni} = 0, \tag{E.3c}$$

where $\hat{y}_0 = y_0$ and $\hat{z}_0 = z_0$. In eq. (E.3), an over-hat has been appended to the symbols of (y_n, z_n) and (Y_{ni}, Z_{ni}) for the purpose of distinguishing them from those of the unperturbed scheme (E.2) within the theorem's proof and the subsequent lemma. Perturbations δ_{n+1}, δ_{ni}, and θ_{ni} in eq. (E.3) correspond to the perturbations in the original IRK-CP scheme (3.3) through eq. (3.12); all of these perturbations are of order $O(h^{\kappa})$, as posited in the theorem. From Theorem 3.1 of [12] and the condition of $\kappa \geq \zeta + 1$, the solution (y_n, z_n) obtained from the IRK-DAE1 scheme (E.2) fulfills eqs. (A.1) and (A.2). In order to apply Lemma 6 from [56], it is assumed that the disparities between (y_n, z_n) and $(\hat{y}_n, \hat{z}_n)$ are bounded by

$$\left\|\hat{y}_n - y_n\right\| \leq C_0 h, \quad \left\|\hat{z}_n - z_n\right\| \leq C_1 h, \tag{E.4}$$

which will subsequently be justified. Additionally, the following notation is introduced

$$\|\delta\| = \max\left(\left\|\delta_{ni}\right\|, \left\|\delta_{n+1}\right\|\right), \quad \|\theta\| = \max\left\|\theta_{ni}\right\|. \tag{E.5}$$

The estimates of eqs. (A.1), (A.2) and (E.4) guarantee the existence of the solution for the perturbed scheme (E.3) in accordance with Lemma 5 from [56]. Consequently, applying Lemma 6 from [56] results in

$$\begin{aligned}
&\left\|\Delta Y_{ni}\right\| \leq C\left(\left\|\Delta y_n\right\| + \|\delta\| + h\,\|\theta\|\right),\\
&\left\|\Delta Z_{ni}\right\| \leq C\left(\left\|\Delta y_n\right\| + \|\delta\| + \|\theta\|\right),
\end{aligned} \tag{E.6}$$

where $\Delta Y_{ni} = \hat{Y}_{ni} - Y_{ni}$, $\Delta Z_{ni} = \hat{Z}_{ni} - Z_{ni}$, and $\Delta y_n = \hat{y}_n - y_n$. By subtracting eq. (E.2a) from eq. (E.3a), the following is obtained

$$\Delta y_{n+1} = \Delta y_n + h \sum_{i=1}^{s} b_i \left[f(\hat{Y}_{ni}, \hat{Z}_{ni}) - f(Y_{ni}, Z_{ni}) \right] + h\delta_{n+1}. \tag{E.7}$$

From eqs. (E.2b) and (E.3b), $\Delta z_{n+1} = \hat{z}_{n+1} - z_{n+1}$ can be expressed in terms of Δz_n and ΔZ_{ni} as follows

$$\Delta z_{n+1} = R(\infty)\Delta z_n + \sum_{i,j=1}^{s} b_i \omega_{ij} \Delta Z_{nj}. \tag{E.8}$$

Incorporating eq. (E.6) into both eqs. (E.7) and (E.8) and employing a Lipschitz condition for the function $f(y,z)$ results in the following recurrence relations

$$\left\|\Delta y_{n+1}\right\| \le \left\|\Delta y_n\right\| + Ch\big(\left\|\Delta y_n\right\| + \|\delta\| + \|\theta\| \big), \tag{E.9a}$$

$$\left\|\Delta z_{n+1}\right\| \le \lambda \left\|\Delta z_n\right\| + C\big(\left\|\Delta y_n\right\| + \|\delta\| + \|\theta\| \big), \tag{E.9b}$$

where $\lambda = |R(\infty)|$. By deduction, eq. (E.9a) implies

$$\left\|\Delta y_n\right\| \le (1+Ch)^n \left\|\Delta y_0\right\| + \sum_{i=1}^{n} (1+Ch)^{i-1} \cdot Ch \left(\|\delta\| + \|\theta\|\right). \tag{E.10}$$

Given that $\Delta y_0 = 0$ and $(1+Ch)^n = 1 + O(h)$ (for $nh \le \mathrm{Const}$), we deduce from eq. (E.10) that

$$\left\|\Delta y_n\right\| \le C \left(\|\delta\| + \|\theta\|\right). \tag{E.11}$$

Substituting eq. (E.11) into eq. (E.9b) yields

$$\left\|\Delta z_{n+1}\right\| \le \lambda \left\|\Delta z_n\right\| + C \left(\|\delta\| + \|\theta\|\right). \tag{E.12}$$

By using a technique similar to that of eq. (E.10), we arrive at

$$\left\|\Delta z_n\right\| \le \begin{cases} C\left(\|\delta\| + \|\theta\|\right) & \text{if } 0 \le |R(\infty)| < 1, \\ C\left(\|\delta\| + \|\theta\|\right)/h & \text{if } |R(\infty)| = 1. \end{cases} \tag{E.13}$$

Now, we verify the assumption of eq. (E.4). It is straightforward to verify that the constants in eq. (E.6) can be chosen independently of C_0 or C_1 if we restrict the step size h such that $C_0 h$ and $C_1 h$ are bounded by h-independent constants. Thus, the constants in eqs. (E.9) to (E.13) do not depend on C_0 or C_1 either. The assumption eq. (E.4) then follows from eqs. (E.5), (E.11) and (E.13) provided that $\kappa \ge 2$ and that C_0 and C_1 are chosen sufficiently large.

From eqs. (A.1), (A.2), (E.11) and (E.13), and the assumption that both $\|\delta\|$ and $\|\theta\|$ are of order $O(h^\kappa)$, it is straightforward to conclude that the global errors given by the IRK-CP scheme (3.3) satisfy eq. (3.14). The convergence of the algebraic-component for the special case in which $R(\infty) = -1$ with the additional assumption of eq. (3.15) can be derived from an asymptotic expansion concerning $\hat{z}_n - z(t_n)$, which is discussed in the following lemma. Finally, for IRK methods of type II, it is easy to prove that the estimate of eq. (E.11) still holds so that $\hat{y}_n - y(t_n) = O(h^\kappa)$. Since $\hat{z}_{n+1} = \hat{Z}_{ns}$ with $\hat{Z}_{ns}$ obtained from the second relation of eq. (E.3c) in this case, we also have $\hat{z}_n - z(t_n) = O(h^\kappa)$ based on the boundedness of $(g_z(y,z))^{-1}$. □

Lemma 4. *Consider a general semi-explicit index-1 system (E.1) and let $(\hat{y}_n, \hat{z}_n)$ denote the numerical solution given by a perturbed IRK scheme (E.3) with the perturbations admitting asymptotic expansions of eq. (3.15). Suppose that its coefficients are of type I, define a standard Runge-Kutta method of order κ for ODEs and satisfy $C(\zeta)$ with $\kappa \ge \max(\zeta, 1) + 1$.*

(a) If $-1 \le R(\infty) < 1$, the global error of the z-component possesses an asymptotic expansion

$$\hat{z}_n - z(t_n) = h^{\zeta+1} \left(b_{\zeta+1}(t_n) + \beta_{n,\zeta+1} \right) + \cdots + h^N \left(b_N(t_n) + \beta_{n,N} \right) + O(h^{N+1}), \tag{E.14}$$

with the h-independent perturbations $\beta_{n,k}$ satisfying

$$\beta_{n,k} = \begin{cases} 0 & \textit{for } n \ge 1, \textit{ when } R(\infty) = 0, \\ O(\alpha^n) & \textit{for all } n \textit{ and any } \alpha \textit{ strictly between } |R(\infty)| \textit{ and } 1, \textit{ when } 0 < |R(\infty)| < 1, \\ (-1)^{n+1} b_k(0) & \textit{for } n \ge 0, \textit{ when } R(\infty) = -1. \end{cases} \tag{E.15}$$

(b) If $R(\infty) = 1$, then

$$\hat{z}_n - z(t_n) = h^\zeta b_\zeta(t_n) + \cdots + h^N b_N(t_n) + O(h^{N+1}). \tag{E.16}$$

The above expansions hold uniformly for $h \le h_0$ and $nh \le Const$.

Remark 2. Lemma 2, which concerns the asymptotic expansion for the global error of the z-component in the IRK-CP scheme, is derived by slightly adapting Theorem 3.2 in [12]. The asymptotic expansion for the global error of the z-component in the perturbed scheme (E.3) is identical in appearance to that of the unperturbed scheme (E.2) (see Theorem 3.2 of [12]). However, its asymptotic coefficients now depend on the asymptotic coefficients of δ_{n+1}, δ_{ni}, and θ_{ni} in eq. (3.15).

Proof. The proof of this lemma can be obtained followings the lines of Theorem 4.8 in [12], which is divided into three parts: In part (a), truncated expansions with exponentially decaying perturbation terms of the forms

$$\begin{aligned}\tilde{y}_n &= y(t_n) + h^{\zeta+1} a_{\zeta+1}(t_n) + \cdots + h^N a_N(t_n),\\ \tilde{Y}_{ni} &= y(t_n + c_i h) + h^{\zeta+1} A_{i,\zeta+1}(t_n) + \cdots + h^N A_{i,N}(t_n),\\ \tilde{Z}_{ni} &= z(t_n + c_i h) + h^{\zeta+1} B_{i,\zeta+1}(t_n) + \cdots + h^N B_{i,N}(t_n),\end{aligned} \tag{E.17}$$

will be recursively constructed such that $\tilde{y}_0 = y_0$ and that

$$\begin{aligned}\tilde{y}_{n+1} &= \tilde{y}_n + h\sum_{i=1}^{s} b_i f(\tilde{Y}_{ni}, \tilde{Z}_{ni}) + h\delta_{n+1} + O(h^{N+2}),\\ \tilde{Y}_{ni} &= \tilde{y}_n + h\sum_{j=1}^{s} a_{ij} f(\tilde{Y}_{nj}, \tilde{Z}_{nj}) + \delta_{ni} + O(h^{N+1}),\\ 0 &= g(\tilde{Y}_{ni}, \tilde{Z}_{ni}) + \theta_{ni} + O(h^{N+1}).\end{aligned} \tag{E.18}$$

In part (b), an equality regarding the $\hat{Z}_{ni}$ in the perturbed scheme (E.3) that has the form

$$\hat{Z}_{ni} = z(t_n + c_i h) + h^{\zeta+1} B_{i,\zeta+1}(t_n) + \cdots + h^N B_{i,N}(t_n) + O(h^{N+1}), \tag{E.19}$$

is proven using eq. (E.13) and Lemma 5 in [56]. In part (c), the difference $\Delta\hat{z}_n = \hat{z}_n - z(t_n)$ is shown to satisfy the recurrence relation

$$\Delta\hat{z}_{n+1} = R(\infty)\Delta\hat{z}_n + \tau_{n+1}, \tag{E.20a}$$

where τ_{n+1} admits an asymptotic expansion

$$\tau_{n+1} = h^{\zeta+1} d_{\zeta+1}(t_{n+1}) + \cdots + h^N d_N(t_{n+1}) + O(h^{N+1}). \tag{E.20b}$$

Finally, the statement of the lemma follows in exactly the same manner as in the proof of Theorem 3.2 of [12].

(a) It can be verified that the defect of $\tilde{y}_n$, $\tilde{Y}_{ni}$, and $\tilde{Z}_{ni}$ defined by eq. (E.17) has the form

$$\begin{aligned}\tilde{y}_{n+1} &= \tilde{y}_n + h\sum_{j=1}^{s} b_i f(\tilde{Y}_{ni}, \tilde{Z}_{ni})\\ &\quad + h^{\zeta+2}\left[a'_{\zeta+1}(t_n) + e_{\zeta+2}(t_n) - f_y(t_n)\textstyle\sum_{i=1}^{s} b_i A_{i,\zeta+1}(t_n) - f_z(t_n)\sum_{i=1}^{s} b_i B_{i,\zeta+1}(t_n)\right] + O(h^{\zeta+3}),\\ \tilde{Y}_{ni} &= \tilde{y}_n + h\sum_{j=1}^{s} a_{ij} f(\tilde{Y}_{nj}, \tilde{Z}_{nj}) + h^{\zeta+1}\left[e_{i,\zeta+1}(t_n) + A_{i,\zeta+1}(t_n) - a_{\zeta+1}(t_n)\right] + O(h^{\zeta+2}),\\ 0 &= g(\tilde{Y}_{ni}, \tilde{Z}_{ni}) - h^{\zeta+1}\left[g_y(t_n)A_{i,\zeta+1}(t_n) + g_z(t_n)B_{i,\zeta+1}(t_n)\right] + O(h^{\zeta+2}),\end{aligned} \tag{E.21}$$

in which the functions $e_k(t)$ and $e_{i,k}(t)$ are respectively given by

$$e_k(t) = \frac{y^{(k)}(t)}{(k-1)!}\left(\frac{1}{k} - \sum_{i=1}^{s} b_i c_i^{k-1}\right), \quad e_{i,k}(t) = \frac{y^{(k)}(t)}{(k-1)!}\left(\frac{c_i^k}{k} - \sum_{j=1}^{s} a_{ij} c_j^{k-1}\right). \tag{E.22}$$

Comparing eq. (E.21) to eq. (E.18) and requiring eq. (E.18) to be satisfied with $N = \zeta + 1$, this yields an index-1 differential algebraic system of the form

$$\begin{aligned}a'_{\zeta+1}(t) &= f_y(t)\textstyle\sum_{i=1}^{s} b_i A_{i,\zeta+1}(t) + f_z(t)\sum_{i=1}^{s} b_i B_{i,\zeta+1}(t) + c_{\zeta+1}(t) - e_{\zeta+2}(t),\\ 0 &= e_{i,\zeta+1}(t) + A_{i,\zeta+1}(t) - a_{\zeta+1}(t) + C_{i,\zeta+1}(t),\\ 0 &= g_y(t)A_{i,\zeta+1}(t) + g_z(t)B_{i,\zeta+1}(t) - D_{i,\zeta+1}(t),\end{aligned} \tag{E.23}$$

where $c_{\zeta+1}(t)$, $C_{i,\zeta+1}(t)$ and $D_{i,\zeta+1}(t)$ equal to zero if $\kappa > \zeta + 1$. Based on the assumption $y_0 = \tilde{y}_0$, the expansion coefficients $a_{\zeta+1}(t)$, $A_{i,\zeta+1}(t)$, and $B_{i,\zeta+1}(t)$ can be uniquely determined from eq. (E.23), and their dependence on the asymptotic coefficients of δ_{n+1}, δ_{ni}, and θ_{ni} (i.e., $c_{\zeta+1}$, $C_{i,\zeta+1}$, and $D_{i,\zeta+1}$) can be noted. Moreover, such a procedure can be recursively repeated with ζ replaced by $\zeta+1, \ldots, N$, ultimately leading to eq. (E.18)

(b) The difference between $\hat{y}_n$ in the system (E.3) and $\tilde{y}_n$ in the system (E.18) can be shown to satisfy

$$\hat{y}_n - \tilde{y}_n = O(h^{N+1}), \tag{E.24}$$

in line with the proof of eq. (E.11) in Theorem 1. Meanwhile, the difference between $\hat{Z}_{ni}$ in system (E.3) and $\tilde{Z}_{ni}$ in system (E.18) can be estimated by applying Lemma 6 of [56] to these two systems and using the estimate eq. (E.24), which admits the form of

$$\hat{Z}_{ni} - \tilde{Z}_{ni} = O(h^{N+1}). \tag{E.25}$$

The equality eq. (E.19) then follows from eq. (E.25) and the third relation of eq. (E.17).

(c) Considering the employed RK coefficients are of order κ for ODEs and satisfy $C(\zeta)$, the exact solution of eq. (E.1) admits the form

$$z(t_{n+1}) = z(t_n) + h\sum_{i=1}^{s} b_i z'(t_n + c_i h) + O(h^{\kappa+1}), \tag{E.26a}$$

$$z(t_n + c_i h) = z(t_n) + h\sum_{j=1}^{s} a_{ij} z'(t_n + c_j h) + O(h^{\zeta+1}). \tag{E.26b}$$

Calculating $z'(t_n + c_i h)$ from eq. (E.26b) and inserting it into eq. (E.26a), this leads to

$$z(t_{n+1}) = R(\infty) z(t_n) + \sum_{i,j=1}^{s} b_i \omega_{ij} z(t_{nj}) + O(h^{\zeta+1}) + O(h^{\kappa+1}), \tag{E.27}$$

Considering $\kappa > \zeta$ and subtracting eq. (E.27) from eq. (E.3b), we arrive at

$$\Delta\hat{z}_{n+1} = R(\infty)\Delta\hat{z}_n + \sum_{i,j=1}^{s} b_i \omega_{ij}\left(\hat{Z}_{nj} - z(t_{nj})\right) + O(h^{\zeta+1}). \tag{E.28}$$

Inserting eq. (E.19) into eq. (E.28), we finally obtain eq. (E.20), which verifies the statement of the lemma following the proof of Theorem 3.2 in [12]. □

Appendix F. A momentum interpolation scheme for the finite volume formulation of the incompressible N-S equations on collocated grids

In this study, the finite volume formulation of the incompressible Navier-Stokes equations on collocated grids is transformed into a system of index-2 DAEs by developing a semi-discrete equation for the momentum (or cell-face velocity) at the cell faces. This is achieved through a custom interpolation of the semi-discrete equation for the cell-centered velocity, which is introduced as follows:

Let's define the notation $A(u)$ and $H(u) \in \mathbb{R}^{m\times m}$ as

$$A(u) = \mathrm{diag}\left(-C(u(t)) + \upsilon D\right), \quad H(u) = \mathrm{offdiag}\left(-C(u(t)) + \upsilon D\right), \tag{F.1}$$

the semi-discrete equation for the cell-centered velocity field $u(t)$, i.e., eq. (1.2a), can then be rewritten as

$$u'(t) = A(u)u + H(u)u - Gp(t) + r(t, v(t)). \tag{F.2}$$

Assuming that the spatial interpolation from a cell-centered vector field $\varphi \in \mathbb{R}^m$ to a face-centered vector field $\bar{\varphi} \in \mathbb{R}^l$ is given by

$$\bar{\varphi} = L\varphi, \tag{F.3}$$

where $L \in \mathbb{R}^{l\times m}$ is the matrix representation of the interpolation operator, we can establish a semi-discrete equation for the face-centered velocity field $\bar{u}(t)$ by a custom interpolation of the first and second terms on the right-hand side of eq. (F.2). This gives

$$\bar{u}'(t) = (LA(u)\mathbb{1}) \odot \bar{u} + LH(u)u - \bar{G}p(t) + \bar{r}(t, v(t)), \tag{F.4}$$

where $\odot$ denotes the Hadamard Product, $\mathbb{1}$ represents a vector of ones, $\bar{G} \in \mathbb{R}^{l\times m}$ is the matrix representation of the gradient operator that computes the gradient of a cell-centered scalar field at the cell faces, and $\bar{r}$ is the counterpart of r in eq. (1.2a). The proposed interpolation scheme differs from the well-known Rhie and Chow interpolation [57], as well as subsequent modifications [58–62], in that it applies the custom interpolation to the semi-discrete momentum equation rather than the fully discrete one. Based on the face-centered velocity field $\bar{u}(t)$, the spatially discretized continuity equation can be written as

$$\bar{M}\bar{u}(t) + Nv(t) = 0, \tag{F.5}$$

where $\bar{M}$ represents the divergence operator matrix applied to $\bar{u}(t)$. Equations (F.2), (F.4) and (F.5) form a system of index-2 DAEs, assuming the existence and boundedness of the inverse of $\bar{M}\bar{G}$. Projecting both sides of eq. (F.4) onto the cell faces yields a semi-discrete equation for the velocity-flux field $\phi(t)$. A continuity equation similar to eq. (F.5) can also be constructed for $\phi(t)$. On stationary grids, constructing the semi-discrete incompressible Navier-Stokes system using $\phi(t)$ instead of $\bar{u}(t)$ offers memory savings.

Data availability

The code and data underlying the numerical experiments of this study are available in

Replication code and data: On convergence of implicit Runge-Kutta methods for the incompressible Navier-Stokes equations with unsteady inflow (Original data) (Mendeley Data)

References

[1] X. Wu, Inflow turbulence generation methods, Annu. Rev. Fluid Mech. 49 (1) (2017) 23–49, https://doi.org/10.1146/annurev-fluid-010816-060322.

[2] L. Patruno, S. de Miranda, Unsteady inflow conditions: a variationally based solution to the insurgence of pressure fluctuations, Comput. Methods Appl. Mech. Eng. 363 (2020) 112894, https://doi.org/10.1016/j.cma.2020.112894.

[3] A. Montorfano, F. Piscaglia, G. Ferrari, Inlet boundary conditions for incompressible LES: a comparative study, Math. Comput. Model. 57 (7) (2013) 1640–1647, https://doi.org/10.1016/j.mcm.2011.10.077.

[4] B. Yan, Q. Li, Inflow turbulence generation methods with large eddy simulation for wind effects on tall buildings, Comput. Fluids 116 (2015) 158–175, https://doi.org/10.1016/j.compfluid.2015.04.020.

[5] G. Lamberti, C. García-Sánchez, J. Sousa, C. Gorlé, Optimizing turbulent inflow conditions for large-eddy simulations of the atmospheric boundary layer, J. Wind Eng. Ind. Aerodyn. 177 (2018) 32–44, https://doi.org/10.1016/j.jweia.2018.04.004.

[6] A.F. Melaku, G.T. Bitsuamlak, A divergence-free inflow turbulence generator using spectral representation method for large-eddy simulation of ABL flows, J. Wind Eng. Ind. Aerodyn. 212 (2021) 104580, https://doi.org/10.1016/j.jweia.2021.104580.

[7] H. Plischka, S. Michel, J. Turnow, B. Leitl, N. Kornev, Comparison of turbulent inflow conditions for neutral stratified atmospheric boundary layer flow, J. Wind Eng. Ind. Aerodyn. 230 (2022) 105145, https://doi.org/10.1016/j.jweia.2022.105145.

[8] X. Wang, C. Cai, P. Yuan, G. Xu, C. Sun, An efficient and accurate DSRFG method via nonuniform energy spectra discretization, Eng. Struct. 298 (2024) 117014, https://doi.org/10.1016/j.engstruct.2023.117014.

[9] L. Chen, C. Li, J. Wang, G. Hu, Y. Xiao, A coherence-improved and mass-balanced inflow turbulence generation method for large eddy simulation, J. Comput. Phys. 498 (2024) 112706, https://doi.org/10.1016/j.jcp.2023.112706.

[10] M.P. Joel H. Ferziger, Computational Methods for Fluid Dynamics, 3rd edition, Springer, Berlin, Heidelberg, 1996.

[11] B. Sanderse, B. Koren, Accuracy analysis of explicit Runge-Kutta methods applied to the incompressible Navier-Stokes equations, J. Comput. Phys. 231 (8) (2012) 3041–3063, https://doi.org/10.1016/j.jcp.2011.11.028.

[12] E. Hairer, C. Lubich, M. Roche, The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods, 1st edition, Springer, USA, 1989.

[13] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II Stiff and Differential-Algebraic Problems, 2nd edition, Springer, USA, 1996.

[14] L.R. Petzold, Order results for implicit Runge-Kutta methods applied to differential/algebraic systems, SIAM J. Numer. Anal. 23 (4) (1986) 837–852, https://doi.org/10.1137/0723054.

[15] M. Roche, Implicit Runge–Kutta methods for differential algebraic equations, SIAM J. Numer. Anal. 26 (4) (1989) 963–975, https://doi.org/10.1137/0726053.

[16] V. Brasey, A half-explicit Runge-Kutta method of order 5 for solving constrained mechanical systems, Computing 48 (2) (1992) 191–201, https://doi.org/10.1007/BF02310533.

[17] V. Brasey, E. Hairer, Half-explicit Runge-Kutta methods for differential-algebraic systems of index 2, SIAM J. Numer. Anal. 30 (2) (1993) 538–552, https://doi.org/10.1137/0730025.

[18] C.W. Gear, B. Leimkuhler, G.K. Gupta, Automatic integration of Euler-Lagrange equations with constraints, J. Comput. Appl. Math. 12 (1985) 77–90, https://doi.org/10.1016/0377-0427(85)90008-1.

[19] P. Lötstedt, L. Petzold, Numerical solution of nonlinear differential equations with algebraic constraints. I. Convergence results for backward differentiation formulas, Math. Comput. 46 (174) (1986) 491–516, https://doi.org/10.2307/2007989.

[20] K.E. Brenan, B.E. Engquist, Backward differentiation approximations of nonlinear differential/algebraic systems, Math. Comput. 51 (184) (1988) 659–676, https://doi.org/10.1090/S0025-5718-1988-0930221-3.

[21] U.M. Ascher, L.R. Petzold, Projected implicit Runge–Kutta methods for differential-algebraic equations, SIAM J. Numer. Anal. 28 (4) (1991) 1097–1120, https://doi.org/10.1137/0728059.

[22] C. Lubich, On projected Runge-Kutta methods for differential-algebraic equations, BIT Numer. Math. 31 (3) (1991) 545–550, https://doi.org/10.1007/BF01933267.

[23] L.O. Jay, Specialized Runge-Kutta methods for index 2 differential-algebraic equations, Math. Comput. 75 (254) (2005) 641–654, https://doi.org/10.1090/S0025-5718-05-01809-0.

[24] A. Murua, Partitioned half-explicit Runge-Kutta methods for differential-algebraic systems of index 2, Computing 59 (1) (1997) 43–61, https://doi.org/10.1007/BF02684403.

[25] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations, J. Comput. Phys. 59 (2) (1985) 308–323, https://doi.org/10.1016/0021-9991(85)90148-2.

[26] H. Le, P. Moin, An improvement of fractional step methods for the incompressible Navier-Stokes equations, J. Comput. Phys. 92 (2) (1991) 369–379, https://doi.org/10.1016/0021-9991(91)90215-7.

[27] U.M. Ascher, S.J. Ruuth, R.J. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, Appl. Numer. Math. 25 (2–3) (1997) 151–167, https://doi.org/10.1016/S0168-9274(97)00056-1.

[28] N. Nikitin, Third-order-accurate semi-implicit Runge-Kutta scheme for incompressible Navier-Stokes equations, Int. J. Numer. Methods Fluids 51 (2) (2006) 221–233, https://doi.org/10.1002/fld.1122.

[29] A. Montlaur, S. Fernandez-Mendez, A. Huerta, High-order implicit time integration for unsteady incompressible flows, Int. J. Numer. Methods Fluids 70 (5) (2012) 603–626, https://doi.org/10.1002/fld.2703.

[30] B. Sanderse, Energy-conserving Runge-Kutta methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 233 (2013) 100–131, https://doi.org/10.1016/j.jcp.2012.07.039.

[31] V. Kazemi-Kamyab, A. van Zuijlen, H. Bijl, Analysis and application of high order implicit Runge-Kutta schemes to collocated finite volume discretization of the incompressible Navier–Stokes equations, Comput. Fluids 108 (2015) 107–115, https://doi.org/10.1016/j.compfluid.2014.11.025.

[32] R. Alexander, Diagonally implicit Runge–Kutta methods for stiff O.D.E.'s, SIAM J. Numer. Anal. 14 (6) (1977) 1006–1021, https://doi.org/10.1137/0714068.

[33] L. Jay, Convergence of a class of Runge-Kutta methods for differential-algebraic systems of index 2, BIT Numer. Math. 33 (1) (1993) 137–150, https://doi.org/10.1007/BF01990349.

[34] I. Higueras, T. Roldán, Starting algorithms for a class of Runge-Kutta methods for index-2 DAEs, Comput. Math. Appl. 49 (7) (2005) 1081–1099, https://doi.org/10.1016/j.camwa.2004.09.006.

[35] F. Cameron, M. Palmroth, R. Piché, Quasi stage order conditions for SDIRK methods, Appl. Numer. Math. 42 (1) (2002) 61–75, https://doi.org/10.1016/S0168-9274(01)00142-8.

[36] L.M. Skvortsov, Diagonally implicit Runge-Kutta methods for differential algebraic equations of indices two and three, Comput. Math. Math. Phys. 50 (6) (2010) 993–1005, https://doi.org/10.1134/S0965542510060072.

[37] L.M. Skvortsov, Third- and fourth-order ESDIRK methods for stiff and differential-algebraic problems, Comput. Math. Math. Phys. 62 (5) (2022) 766–783, https://doi.org/10.1134/S0965542522050128.

[38] J.C. Butcher, Implicit Runge-Kutta processes, Math. Comput. 18 (85) (1964) 50–64, https://doi.org/10.1090/S0025-5718-1964-0159424-9.

[39] L.M. Skvortsov, Diagonally implicit Runge-Kutta FSAL methods for stiff and differential-algebraic systems, Mat. Model. 14 (2002) 3–17 (in Russian).

[40] M.H. Carpenter, D. Gottlieb, S. Abarbanel, W.-S. Don, The theoretical accuracy of Runge-Kutta time discretizations for the initial boundary value problem: a study of the boundary error, SIAM J. Sci. Comput. 16 (6) (1995) 1241–1252, https://doi.org/10.1137/0916072.

[41] A. Biswas, D.I. Ketcheson, B. Seibold, D. Shirokoff, Design of DIRK schemes with high weak stage order, Commun. Appl. Math. Comput. Sci. 18 (1) (2023) 1–28, https://doi.org/10.2140/camcos.2023.18.1.

[42] R.R. Rosales, B. Seibold, D. Shirokoff, D. Zhou, Spatial manifestations of order reduction in Runge-Kutta methods for initial boundary value problems, arXiv:1712.00897, 2023.

[43] A. Biswas, D.I. Ketcheson, S. Roberts, B. Seibold, D. Shirokoff, Explicit Runge Kutta methods that alleviate order reduction, arXiv:2310.02817, 2023.

[44] S. Abarbanel, D. Gottlieb, M.H. Carpenter, On the removal of boundary errors caused by Runge–Kutta integration of nonlinear partial differential equations, SIAM J. Sci. Comput. 17 (3) (1996) 777–782, https://doi.org/10.1137/S1064827595282520.

[45] D. Pathria, The correct formulation of intermediate boundary conditions for Runge–Kutta time integration of initial boundary value problems, SIAM J. Sci. Comput. 18 (5) (1997) 1255–1266, https://doi.org/10.1137/S1064827594273948.

[46] I. Alonso-Mallo, Runge-Kutta methods without order reduction for linear initial boundary value problems, Numer. Math. 91 (2002) 577–603, https://doi.org/10.1007/s002110100332.

[47] R.I. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, J. Comput. Phys. 62 (1) (1986) 40–65, https://doi.org/10.1016/0021-9991(86)90099-9.

[48] R. Santos, L. Alves, A comparative analysis of explicit, IMEX and implicit strong stability preserving Runge-Kutta schemes, Appl. Numer. Math. 159 (2021) 204–220, https://doi.org/10.1016/j.apnum.2020.09.007.

[49] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (2) (1988) 439–471, https://doi.org/10.1016/0021-9991(88)90177-5.

[50] D.I. Ketcheson, C.B. Macdonald, S. Gottlieb, Optimal implicit strong stability preserving Runge–Kutta methods, Appl. Numer. Math. 59 (2) (2009) 373–392, https://doi.org/10.1016/j.apnum.2008.03.034.

[51] G.G. Dahlquist, A special stability problem for linear multistep methods, BIT Numer. Math. 3 (1) (1963) 27–43, https://doi.org/10.1007/BF01963532.

[52] B.L. Ehle, A-stable methods and Padé approximations to the exponential, SIAM J. Math. Anal. 4 (4) (1973) 671–680, https://doi.org/10.1137/0504057.

[53] L.M. Skvortsov, Diagonally implicit Runge-Kutta methods for stiff problems, Comput. Math. Math. Phys. 46 (12) (2002) 2110–2123, https://doi.org/10.1134/S0965542506120098.

[54] J. Mann, Wind field simulation, Probab. Eng. Mech. 13 (4) (1998) 269–282, https://doi.org/10.1016/S0266-8920(97)00036-2.

[55] G. Comte-Bellot, S. Corrsin, Simple Eulerian time correlation of full- and narrow-band velocity signals in grid-generated isotropic turbulence, J. Fluid Mech. 48 (2) (1971) 273–337, https://doi.org/10.1017/S0022112071001599.

[56] E. Hairer, C. Lubich, M. Roche, Error of Runge-Kutta methods for stiff problems studied via differential algebraic equations, BIT Numer. Math. 28 (3) (1988) 678–700, https://doi.org/10.1007/BF01941143.

[57] C.M. Rhie, W.L. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation, AIAA J. 21 (1983) 1525–1532, https://doi.org/10.2514/3.8284.

[58] S.K. Choi, Note on the use of momentum interpolation method for unsteady flows, Numer. Heat Transf., Part A, Appl. 36 (5) (1999) 545–550, https://doi.org/10.1080/104077899274679.

[59] B. Yu, W.-Q. Tao, J.-J. Wei, Y. Kawaguchi, T. Tagawa, H. Ozoe, Discussion on momentum interpolation method for collocated grids of incompressible flow, Numer. Heat Transf., Part B, Fundam. 42 (2) (2002) 141–166, https://doi.org/10.1080/10407790190053879.

[60] A. Pascau, Cell face velocity alternatives in a structured colocated grid for the unsteady Navier–Stokes equations, Int. J. Numer. Methods Fluids 65 (7) (2011) 812–833, https://doi.org/10.1002/fld.2215.

[61] D.K. Kolmogorov, W.Z. Shen, N.N. Sørensen, J.N. Sørensen, Fully consistent CFD methods for incompressible flow computations, J. Phys. Conf. Ser. 524 (2014) 012128, https://doi.org/10.1088/1742-6596/524/1/012128.

[62] P. Bartholomew, F. Denner, M.H. Abdol-Azis, A. Marquis, B.G. van Wachem, Unified formulation of the momentum-weighted interpolation for collocated variable arrangements, J. Comput. Phys. 375 (2018) 177–208, https://doi.org/10.1016/j.jcp.2018.08.030.